UNF UNIVERSITY of NORTH FLORIDA

UNF Digital Commons

UNF Graduate Theses and Dissertations | Student Scholarship

2003

dbUNiFier: A Framework for Automated Unification of Textual Data in Multiple Remote Data Sources

John S. Dembowski
University of North Florida

Suggested Citation

Dembowski, John S., "dbUNiFier: A Framework for Automated Unification of Textual Data in Multiple Remote Data Sources" (2003). *UNF Graduate Theses and Dissertations*. 366.
https://digitalcommons.unf.edu/etd/366

This Master's Thesis is brought to you for free and open access by the Student Scholarship at UNF Digital Commons. It has been accepted for inclusion in UNF Graduate Theses and Dissertations by an authorized administrator of UNF Digital Commons. For more information, please contact Digital Projects.
© 2003 All Rights Reserved

UNF UNIVERSITY of NORTH FLORIDA

dbUNiFier: A Framework for Automated Unification of Textual Data in Multiple Remote Data Sources

by

John S. Dembowski

A thesis submitted to the
Department of Computer and Information Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES

April 2003

Copyright (©) 2003 by John S. Dembowski

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of John S. Dembowski or designated representative.

The thesis "dbUNiFier: A Framework for Automated Unification of Textual Data in Multiple Remote Data Sources" submitted by John S. Dembowski in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Science has been

Approved by the thesis committee: Date

Signature Deleted

4/23/03

Behrooz K. Seyed-Abbassi
Thesis Adviser and Committee Chairperson

Signature Deleted

23 Apr 2003

Yap S. Chua

Signature Deleted

23 April 2003

Sherif A. Elfayoumy

Accepted for the Department of Computer and Information Sciences:

Signature Deleted

4/29/03

Judith L. Solano
Chairperson of the Department

Accepted for the College of Computing Sciences and Engineering:

Signature Deleted

5/5/03

Neal S. Coulter
Dean of the College

Accepted for the University:

Signature Deleted

5/8/03

Thomas S. Serwatka
Dean of Graduate Studies

ACKNOWLEDGEMENT

I wish to thank Penny Ortega and my family for their support during my long education process. I would also like to thank the University of North Florida for the education and specifically Dr. Behrooz Seyed-Abbassi for pushing me during the past year.

CONTENTS

FIGURES

TABLES

ABSTRACT

Over time, advances in database technology and utilization have resulted in a rapid increase in the number and types of data sources. Simultaneously, numerous methods of unifying these various data sources have emerged. Research has shown that a more comprehensive set of data attribute matches between multiple schemas can be detected by combining a number of the unification methodologies as opposed to using a single method. In this research project, a unification framework, dbUNiFier, has been proposed as an approach to allow for easy integration of both existing and future unification methods and data sources.

Chapter 1

Introduction

Unification of multiple data sources allows for further examination or exploitation of the data contained in those sources. To manually unify the contents of any number of remote data sources with any other data sources, a human must compare schema information for the data sources. A schema is a representation, often graphical, of all relationships found in a data source. Information about entities in the data sources is stored within an attribute of that entity. When two attributes contain the same data, data in one attribute is a subset of another, or the data of two attributes are subsets of the same set, then the two attributes are considered a match. As matches are found, they are added to a new, unified schema that reflects the match. Once all matches are found then the remaining unmatched attribute information is transferred to the new schema. Additional remote data sources can be added by comparing the new data source schema with the unified schema. This is very time consuming and possibly overwhelming if the number of sources is large, the attribute names are cryptic, the source types differ greatly, or the sources lack schemas or metadata about the attributes.

Assuming that a human's time is expensive and a computer's time is inexpensive, it is better to have a computer tackle complex problems that it has the ability to solve or aid in finding the solution. Since remote data source unification cannot be fully automated [Miller01], the goal should be to reduce the amount of human evaluation time while exploiting computer evaluation time. Table 1 shows a hypothetic hourly expense for the unification process. Reducing the amount of human time used and increasing the amount of computer time used can reduce the total expense of a project of this nature.

	Cost per Hour *	Cost per Week **
Human:	$50.00	$2,000.00
Automated:	$0.50	$84.00

* This is an example of Human cost per hour and would include all business expenses to maintain the employee including wage, taxes, facility, etc. Computer cost per hour includes all business expenses to maintain the computer including acquisition, maintenance, support, facility, etc.
** This is an example of Human cost per week based on 40-hour workweek. Computer cost based on 168-hour workweek.

Table 1: Possible Hourly Expense for Unification Process

Extensive research exists which addresses methodologies of automated unification of databases and other data sources. Each method examines the data and compares each attribute with attributes from other sources. Simple methods compare one or a few aspects of the data including meta-data, data items, data descriptions, type, and size. Complex methods

use thesauri or dictionaries to expand the meanings of single words or combinations of words; convert from one language to another; or take into account synonyms, homonyms, or hyponyms. More advanced methods combine the results of several simpler methods [Madhavan01, Rahm01].

Each new data source introduced to the unification process must be integrated into the process to allow for information to be gathered from that data source. Existing applications allow for integration of specific data source types and a few allow for multiple data source types. Most methodologies leave the process of gathering information from the data source to the implementer of the methodology.

Existing evaluation methods take into account the attribute level equality of names, equality of canonical name representation, equality of synonyms, equality of hyponyms, sub strings, pronunciation, soundex, user provided matches, description of attributes, cardinality, schema mappings, data type, length, keys, or other information. Some examine information at a variety of levels of granularity. Some examine instance level information [Rahm01].

As matches are located, the match information (mapping) is either stored in a mapping table [Miller01] or the data is converted to an internal representation and stored locally. Once the evaluation of the data is completed, a human must

verify the findings. Then another attempt to match data can be executed taking the verified mappings into consideration. Once the administrator is satisfied that all mappings have been uncovered, a final mapping table or a full internal representation is created [Clifton97]. When new attributes or data sources are added, the process can be started from scratch or only the new pieces can be compared to the internal schema [Mitra99].

Since no one method will be able to resolve all matches in all situations, a combination of methods is the best approach to automated data unification [Madhavan01]. Each single method requires a finite amount of time to complete its search for matches so each additional method added will increase the total processing time. The trade off may result in more potential matches being found.

1.1 Existing Methodologies

To develop this list, a collection of existing unification methods was gathered. Methods with little or incomplete information were discarded. The remaining methods were then examined looking for similarities in their structure and/or the way they go about the unification process.

1.1.1 Specific Methods

A variety of methods have been developed to unify databases. They have been developed over time with some building upon previous methods and others being unique methods. Unifiers have been developed separately by companies and educational organizations. Many of the methods were designed to evaluate specific databases or solve a specific problem. The sample of unification methods presented are the methods that are best referenced and most widely reviewed and analyzed.

1.1.1.1 VIbE

VIbE (Visual Integration by Example), as the name suggests, relies on human input of existing matches. The first step in using this method is to input a sample schema consisting of known matches. It then uses this information to further evaluate the two data sources. The method is dependent upon Java and the FraQL database querying language [Geist01]. VIbE determines new matches based on the matches input by the user. It looks for exact matches and does not suggest possible matches for anything but exact matches. Conflicts are reported in a color coded scheme: white means no conflict and red signifies a conflict [Geist01].

This method offers numerous views of the internal data and the external sources. The goal with the variety of views is to allow the user to analyze the data as well as entering mappings. "The *import view editor* displays the imported relation together with mapping information" [Geist01].

1.1.1.2 Delta

Delta (Data Element Tool-Based Analysis) is a product of the MITRE Corporation. The method uses "textual similarities between data element definitions to find candidate attribute correspondences" [Clifton97]. The unifier accepts queries from the operator and then examines documents based on the query. The user then looks for potential matches in the results of the query.

When using the Delta method, it helps to have knowledge of the domains. Delta converts meta-data about the attributes into text strings. The strings are then stored as documents in the retrieval system [Rahm01]. The operator selects which attributes to investigate against which databases and sets the search criteria. The search criteria can include wild cards as part of the search [Clifton97]. The tool compares attributes against the document base for target databases. The operator examines the ranked returns to determine which potential matches are truly related [Clifton97]. During external analysis of this method,

fifteen minutes per attribute were needed to perform the unification process and that the tool was easy to use [Clifton97].

1.1.1.3 SemInt

SemInt was developed at Northwestern University. It is a 1:1 matcher [Rahm01]. The SemInt method is useful when domain knowledge and documentation are lacking [Clifton97]. It is an "instance based matcher that associates attributes in the two schema with match signatures" [Madhavan01]. It derives twenty numeric properties from meta-data and the database population. The fifteen constraint-based and the five content-based metrics create a two-dimensional representation of the similarities between the attributes [Madhavan01]. SemInt uses a back-propagation three layer neural network to determine "which properties are most useful for discriminating among attributes and produces classifier functions" [Clifton97]. The method relies heavily on data information: type, length, keys, foreign keys, value, constraints, range constraints, access restrictions. It uses statistics found from the population: average, max, and variances [Clifton97]. Like items are clustered by Euclidean distance [Madhavan01].

Unification with SemInt is a two-step process. First, the neural network is trained to recognize attributes by their

descriptors. Next the neural network is used to map attributes from one database onto attributes from the second database. The results are output as a ranked list of similarities between attributes. It was found that SemInt runs well as a batched process requiring little interaction and that SemInt works best when used to supplement other method(s). The method does not produce good results on its own but may be useful as one of the components of larger unification processes [Clifton97].

1.1.1.4 MOMIS

The Mediator envirOnment for Multiple Information Sources (MOMIS) [Beneventano01] approach uses layers called wrappers to insulate the unification application from the external data sources. The wrapper is defined in the method. A language called ODL is used to translate meta-data of the data source to a common data model (called ODM), translates queries, and exports query results. A layer called Schema Integrator-Designer processes the information obtained from the wrapper. A Query Management Module handles query processing and optimization [Beneventano01].

MOMIS is a tool that incorporates expert knowledge of the domains being unified. To do this it uses the thesaurus built around the knowledge of the domain covered by the databases involved. A typical run of MOMIS first generates

the thesaurus. The thesaurus generation is a semi-automatic process and requires the help of the expert operating the application. It helps define inter- and intra-schema relationships. The thesaurus is used by the unifier to expand upon the meta-data of the attributes involved [Beneventano01]. The application analyzes the affinity of the classes found in the two data sources. The classes are clustered based on any affinity determined with the help of the thesaurus. Lastly it generates the unified schema [Beneventano01].

MOMIS also defines a Mediator that is used to handle the local data. The Mediator is broken into two pieces. An SI-Designer processes information from the wrapper. A QM Module handles query processing and optimization [Beneventano01]. The SI-Designer handles five distinct tasks: source acquisition; intentional relationship definitions; extensional relationship definitions; clustering; and mapping table tuning. The intentional relationship definer adds relationships using numerous methods. The clustering piece uses knowledge in the thesaurus. The mapping table tuner requires user interaction [Beneventano01].

The MOMIS method uses a mapping table to store intentional mappings. Columns represent local classes belonging to a cluster. Rows represent global attributes [Beneventano01].

MOMIS is able to integrate data from structured and semi-structured data sources [Gelati01].

1.1.1.5 Learning Source Descriptions

Learning Source Descriptions (LSD) works on XML documents and uses multi-level learning to perform 1:1 matching. It examines both attribute and instance-level data during the unification process. A meta-learner combines predictions in a weighted manner determined by the meta-learner. [Madhavan01] Training data is constructed from a set of sample mappings. A sample set of instance data is extracted and used. The learner tries to distinguish the attribute characteristics of the mediated data set. Upon completion of training, the software should be able to determine if new instances are members of a specific attributes that has been tested [Doan00].

During the classification phase, the unifier examines the attribute or a set of instance items. It attempts to classify the attribute in question as one of the attributes learned during the learning phase. It combines and reports the findings from each individual [Doan00]. There are four learners used in the defined unifier but the author points out that any learner can be used. A Whirl learner focuses on the attributes neighbors. A Naïve Bayesian learner examines word frequency. A Name matcher examines similarity

of names of synonyms. A Country-Name recognizer uses a database of country names. The Country-Name recognizer is an example of a specialized piece that can be added based on the nature of the data sources being examined. A unifier based on the LSD method using such a specialized piece is then able to tell if an instance is a country name [Doan00]. A fifth learner combines the results of the four sub-learners. LSD combines the sub-learner scores in a way that is relevant to the attribute being examined. If one learner weighs more heavily in the original training phase then that weight is included in its final evaluations [Doan00].

1.1.1.6 Semantic Knowledge Articulation Tool

Semantic Knowledge Articulation Tool (SKAT) uses first order logic. It uses name matching and simple structure matching based on the 'is-a' relationship [Madhavan01]. The user supplies specific items to match and specific items not to match. SKAT uses this expert knowledge to suggest additional matches. The user approves, rejects, or marks as irrelevant. SKAT then updates the rules set and reprocesses the data based upon the both the new and existing rules [Mitra99]. SKAT examines ontologies which are "knowledge structures which explain the nature and essential properties and relations between terms present in a knowledge source" [Mitra99]. To do this it uses an ontology algebra that

examines similarities between the content of the data sources.

SKAT defines four types of mismatches. Term Semantic Mismatches occur when two documents use the same term to reference different concepts. They can also occur when two documents refer to the same concept but use different terminology. SKAT assumes that a single term within an ontology refers to the same concept when used throughout the source [Mitra99]. Structural Mismatches occur when a single term from one source is used to reference multiple terms in another source document which causes one attribute to match multiple attributes in a second document. This type of mismatch is allowed since it is a real world possibility [Mitra99]. Instance Mismatches occur when one source defines or describes an attribute differently than the second source. If this is anticipated, an articulation rule can be added to overcome the mismatch. If the rule does not exist, a match will not be found [Mitra99]. Granularity Mismatches are generated when "in the first graph a concept has been organized into a more elaborate hierarchy than in the second one" [Mitra99]. The operator can choose to match the portions of the graph that match. Otherwise the unifier does not consider the items a match.

SKAT performs several preprocessing steps that are used to identify multiple instances of the same term applying to

multiple concepts within the same source or between sources. The operator also compiles a list of 'stop words' and can extend their meanings by providing their root. Stop words are generally common words such as 'and', 'nor', or 'on'. The user can add additional rules to the preprocessing procedure [Mitra99].

1.1.1.7 TransScm

In this method, the schema is converted to a graph. Multiple matching rules are applied to each node in a fixed order. The process is top-down. A match is determined if one of the internal matching rules is met [Madhavan01]. TransScm is able to determine both element-level and schema level matches. The later is determined by comparing the two graphs. When portions or both graphs are identical, TransScm has found a schema level match. The unifier works well when the top-level of the schemas are similar [Madhavan01].

1.1.1.8 Artemis

Artemis is a schema integrator. It is a hybrid that uses both element and structural information to determine matches. It examines the data sources for similarities and assigns a 0 to 1 rank between the attributes of each source.

It clusters classes to get global classes. MOMIS uses Artemis for its schema integration [Madhavan01].

Artemis uses a generic or domain specific thesauri, and a list of compatible data-types. To determine structural matches, it evaluates relationships that the attributes have amongst attributes in their schemas. It determines a true match by weighting individual component matches, and summing them [Rahm01].

1.1.1.9 Cupid

Cupid was designed and developed by MicroSoft Corporation. Cupid uses linguistics matching at both the element and the structural levels. "It is biased toward similarity of atomic elements" [Madhavan01]. Cupid represents the schemas locally as graphs and uses a tree matching algorithm to determine structural matches. The algorithm can resolve variations in the structure. Cupid takes into account keys, referential constraints, and views. The approach is purely schema-based and not instance-based [Madhavan01]. Cupid allows for depth variations when attempting unification. Since the representation is a graph, the node depth can be taken into account. If a node is past a specified depth, it will not be examined. These deep nodes will be brought into the final unified schema based on their relationship to the deepest node of relevance [Madhavan01].

The operator can mark specific attributes as *optional*. Within a branch of the internal graph, if an optional attribute is present, the leaf node is considered optional as well. The unifier adjusts its scoring based on whether the attribute is implied or implicitly optional [Madhavan01]. This method allows for previously determined mappings to be entered prior to a unification attempt. The mapping table is modified as entries are added during the unification process. Once the new mappings are validated, a subsequent unification run can be initiated which will take into account the newly discovered and validated mappings [Madhavan01].

1.1.1.10 Clio

Clio is not an automated unification method. It is more of a unification manager. "Integration tasks that involve matching often cannot be fully automated since the syntactical representations of schemas, meta-data, and data may not completely convey the semantics of different data sets. As a result, the unifier must rely on an outside source (either a user or a knowledge discovery technique) to provide some information about how different schemas (and data) correspond" [Miller01].

The Clio application is a product of IBM Corporation. It uses a GUI to accept human input to map attributes together.

A small amount of unification occurs by examining existing matches and the matched attributes relation to other attributes in both the original and unified schema [Miller01]. "Clio does not perform schema integration, per se. Rather, Clio supports the generation and management of schemas, correspondences between schemas and mapping (queries) between schemas" [Miller01]. The process is broken into two parts. A correspondence engine generates a set of candidate correspondences between two schemas. Clio uses an attribute classifier to sort through the databases. Correspondences can be modified, accepted, or rejected. A mapping engine then creates and stores mappings. The engine also allows for modification or deletion of existing matches [Miller01].

1.1.2 Grouping by Unification Type

The above methods can be broken into eight groupings. Some of the existing methods are diverse enough that they can be classified in more than one grouping. The idea is not to classify the method but to use the method or the features of the method as an example for a generalized group. When this grouping list was first developed, it consisted of the first seven groups. As additional research was conducted in determining a unification model to emulate for comparison of the proposed method, it became clear that an eighth group for unobtainable methods was needed. These are methods that

exist but for one reason or another are out of reach for this type of research. Although this list may be incomplete or become outdated, it gives a good picture of a variety of unification types. Unification is a multifaceted problem. The problem can be examined and possibly resolved by looking at different pieces of information that can be determined about the information or data found in two data sources.

1.1.2.1 Example-Driven Integration

Example-driven methods rely on known mappings. These mappings are input into the application. The software takes the known mappings and extends them through other known mappings or affinities. While Example-Driven Integration does generate a unified schema, it relies heavily on human time to determine the matches, enter the matches into the system, and evaluate the matches that the application presents. This method does not perform any matching of attributes nor does it allow for addition of this feature. It extends matches by following relationships within the schema. Matches may be missed due to the operator not knowing the data sources thoroughly or the integrator being overwhelmed when integrating large or numerous data sources. The application may miss matches if the schemas are not complete. Clio [Miller01] is an example of this type of unifier. It relies entirely on human input to map attribute matches. The operator uses a GUI and manually maps desired

items between two data sources [Geist01]. The data entry method could be any method including stored files, objects, incoming data stream, etc.

Once the desired mappings have been entered, this method creates a Generic Integration Model. This is an intermediate step to integration. The method detects semantic conflicts and resolves conflicts through designing new schemas and mappings from the Generic Integration Model for the unified schema. The method defines a way for resolving vertical conflicts and another for extensional conflicts. It automatically groups attributes by internal and external conflicts. This portion of the process can be improved by the user adding additional constraint rules [Geist01].

"Given the integrated schema, example-driven integration is the process of creating iteratively and interactively mappings between local schemas and the global schema driven by evaluation of the current data" [Geist01]. Conflict resolution and analysis are performed iteratively. This method allows for manual refinement of the integrated schema [Geist01].

Two other methods using similar processes are VIbE (Visual Integration by Example) and SIGMAbench [Geist01]. Both VIbE and SIGMABench work in essentially the same manner. The

operator inputs known mappings and the application extends the mappings based on relationships found in the schema.

1.1.2.2 Meta-data Examination

Meta-data methods rely on characteristics of the attributes being examined. Meta-data is "the detailed description of the instance data; the format and characteristics of populated instance data; instances and values dependent on the role of the meta-data recipient" [Tannenbaum02]. The author, originator, size, formatting, or other characteristics about date are often included as meta-data [Chapple03]. The additional information helps to give meaning to the data as a whole without having to examine the individual data units. The use of meta-data can allow for more complex searching or comparing to be done. By using the meta-data, large pieces of information can be selected to be excluded or included from a process.

Delta [Clifton97] is an example of a unification technique that relies on meta-data. This particular method converts the meta-data, including definitions, about attributes into text strings. The tool compares the attribute strings against the document base for the target database [Clifton97]. Semantic Integrator (SemInt) [Clifton97] is another example of a meta-data based method. SemInt uses twenty numeric properties that it determines from the meta-

data and the data population. For both Delta and SemInt, the results are ranked in an order of certainty. The integrator examines the results to determine which are truly related [Clifton97]. Delta works best when ample information is available about the attributes. SemInt is valuable when less information is available about the attributes.

1.1.2.3 Textual Similarities of Attributes

Textual similarities can be as simple as comparing the attribute names to see if they are the same, similar, or one is a substring of the other. This is a good way to find a match between 'phone' and 'phone_number'. However, this method may find many matches that on the surface seem probable but the instance data may show otherwise.

One of the learning components of LSD, Whirl, examines attribute names looking for similarities between pairs of names. "It uses statistical measures of document similarity that have been developed in the information retrieval community" [Cohen98]. Vector space representation is used to represent each attribute. An A* Search, a complete and optimally efficient search algorithm [Russell95], is used to find similar attributes. Whirl is used specifically for unstructured data [Cohen98]. However, once the data source

is converted to a local representation it can be used on traditional database data.

1.1.2.4 Structure of Attribute, Data Items, Tables, or Schema

LSD also uses this method and classifies based on neighbors in training sets. LSD uses a multi-strategy machine learning approach to determine characteristics of data and assigns them to the attributes. It can determine that a home address consists of numbers and words; a phone number has ten digits; small numbers indicate a room number while large numbers indicate prices; or that an agency phone number is related to an agency name due to the proximity of the attributes [Doan00].

First, the program goes through a learning phase. Sample mappings are entered. Sample data from the data population are extracted and entered into the program. The learning portion of the software tries to determine what makes the data conform to the mappings so it can predict new data mappings [Doan00]. The method identifies that a multi-strategy approach is more effective than a single approach and allows for extensions when new learners are developed [Doan00]. It works best on verbose and textual data or data which is limited and uniquely indicative [Doan00]. Naïve Bayesian Learner [Doan00], used by Whirl, works best when strong words related to an attribute are not used in other fields. It also performs well in instances of weak words

that are used often. It is not good for short fields or numeric fields [Doan00]. Other methods are used and a meta-learner combines the predictions of each. It uses the training data to determine each learner's importance to that attribute. It then weights each prediction accordingly [Doan00].

DIKE and Artemis also examine the schemas at the structure level.

1.1.2.5 Extended Linguistics

The linguistics methods are very strong because they allow words or phrases to be abstracted into other words or phrases that have a similar meaning. Methods use dictionaries, thesauruses, translations, or other piece to give additional information about words or phrases possibly found in the schema. This abstraction helps to provide additional information that may be used to find matches. The MOMIS method uses information found in a thesaurus to help find matching attributes. The method allows for the semi-automated creation of the thesaurus for the specific schemas being unified. Specialized thesauri work well in translating from language to language, within a specific domain, or possibly across similar domains. LSD, SKAT, TranScm, and DIKE also extend the attributes by examining synonyms, homonyms, or hypernyms [Rahm01].

1.1.2.6 Graphs and Structure Matching

TransScm [Madhavan01] converts the schemata to graphs. Multiple matching rules are applied to each node in a top-down fashion. This method works well when the top-level of schemata are similar [Madhavan01]. Cupid [Madhavan01] uses several methods; including a structure matcher that attempts to match schema trees. It examines leaf nodes for 1:1 relationships. It also determines that two trees are similar if the linguistics of two nodes and their sub trees are similar [Madhavan01].

The DataGuide method does a depth first search through a database and creates a hash table of the information it finds. It keeps track of items previously examined and paths explored. The method compares all objects to see if they are similar to other objects; compares labels or attribute; examines paths to objects and uses this information to find similar items that would not be found in other ways. It also examines node labels along a path to see if a path can be shortened [Goidman99]. "DataGuide is a concise and accurate structural summary of a semi structured database" [Goidman99].

1.1.2.7 Combined Approaches

Existing combined approaches use more than one method to evaluate schemata in an attempt to match attributes. Generally, a scoring system is used to rank the findings and a weighting system is used to skew the findings based on matchers that are more relevant than others. "The problem of schema matching is so hard, and the useful approaches so diverse, that only by combining many approaches can we hope to produce truly robust functionality... Cupid, Dike, and MOMIS are roughly comparable, in that they are purely schema-based and do element- and structure-level matching. Cupid and MOMIS also have a linguistics-based matching component, which are significantly different" [Madhavan01]. Each method evaluates the schema at different levels in an attempt find the most matches possible [Madhavan01].

1.1.2.8 Limited Combined Approaches

A portion of the group of combined approaches can be further broken into a sub-group of methods that for one reason or another cannot be examined completely in their defined state. These methods either rely on a particular platform or language, are in a stage of development, are proprietary, or are theoretical for all practical purposes. These methods are combined methods, but because of their limitations should be identified as a separate grouping.

VIbE relies on the FraQL querying language and a working application is not available to the public. A working MOMIS application is not available to the public. SIGMAbench is underdeveloped and essentially a theoretic unifier.

1.1.3 Attribute-Level and Schema-Level Matching

All of the methods examined determine matching on an attribute level. Most of the methods do not match table to table or schema to schema except by determining that each attribute of the first table corresponds to an attribute in a second table. Attribute- or "schema-level matchers only consider schema information, not instance data" [Rahm01]. Some take into account the fact that a particular attribute is in a specific table and will use this fact to expand the method's abilities. They do not determine structural matches on anything but an attribute level.

At the attribute level, there are several types of matches. A match can be determined by any method. A 1:1 match is between two single attributes. A 1:n match is between a single attribute in one data source and a permutation of attributes from another data source. A m:n match would be multiple attributes being combined to match a permutation of remote attributes. Table 2 shows the various match types.

Type of Match	Evaluation Complexity	Description
1:1	(X * Y)	Match between a single attribute and another single attribute
1:n	X*YPn	Match between a single attribute and the combination of multiple attributes
m:1	XPm*Y	Match between the combination of multiple attributes and a single attribute
m:n	XPm*YPn	Match between the combination of multiple attributes and a separate combination of multiple attributes

Table 2: Types of Matches Between Attributes

All the examined methods work to determine 1:1 matches. This level of matching is the simplest to achieve. "Previous work has mostly concentrated on 1:1 matches because of the difficulty to automatically determine the mapping expressions in the other cases" [Rahm01].

An effective way to measure the work that a method performs is to calculate the number of comparisons needed to execute the method. To fully compare every attribute in one source (X) to every attribute in another source (Y) requires a small amount of comparisons equal to the product of the number of attributes from each source being unified in a specific run or (X * Y). For example assume Source 1 has 50 attributes and Source 2 has 40 attributes. To compare all

attributes using a 1:1 method will require a minimum of 2000 comparisons.

If the method requires multiple comparisons, then the number is increased as a multiple of the number of times a comparison is performed on two distinct attributes to complete that method. If a method uses 15 types of comparisons to complete the method, then a fifteen times multiplier will determine the number of comparisons needed to do a complete comparison of the two data sources under that method or (15 * (X * Y)).

SKAT is able to make 1:n or m:1 level matches. The minimum amount of work to evaluate a single attribute with any number of attributes from a second source is a permutation of Y items in sets of n where Y is equal to the total number of attributes in source two. The number of comparisons needed is then multiplied by the number of attributes in the first source X. The formula would be X*YPn. If Source 1 has 50 attributes and Source 2 has 40 attributes being combined 3 at a time, to compare all attributes using a 1:3 method will require 50*40P3 or 2,964,000 total comparisons if the method uses only one type of comparison. If more than one comparison is needed then the number will be multiplied by the number of comparisons needed Z. The formula is Z*(X*YPn).

Because the numbers increase so quickly, it becomes almost impossible to conduct a meaningful m:n attribute level comparison. The total work can be found as a product of the XPm*YPn. That product is multiplied by the total number of comparisons required by the method Z. If Source 1 has 50 attributes total being combined 3 at a time. Source 2 has 40 attributes total being combined 3 at a time. To compare all groups of attributes between the sources using a 3:3 method will require 6,971,328,000 total comparisons. At .01 second per attribute level evaluation it will require 2.2 years to complete a full evaluation of possible matches between the two data sources at the attribute level. That is the method only requires one comparison between each set of attributes between the two sources. If more than one comparison is needed then the number will be multiplied by the number of comparisons needed Z. The formula is Z*(XPm*YPn).

1.1.4 Instance-Level Matching

Instance level matching takes into account data that is included as an attribute to an item. By their very nature, data source can include hundreds, thousands or even millions of instances for a specific attribute. Any meaningful analysis at the instance-level can involve examining very large amounts of data just to compare one attribute to

another. Methods have been developed to minimize the examination of these items.

1.1.4.1 Neural Nets

Methods that use neural nets are difficult to measure with a single formula. The efficiency of the net depends on the implementation used, the data set size and the number of runs needed to learn the data will depend on the situation and the implementation being used.

1.1.4.2 Comparisons of Instance Data

Again, a generic measurement for methods comparing instance data is difficult. Methods, like Delta, examine the data elements as a whole in the form of a document. Not enough information was available to analyze this type of matching.

1.2 Deficiencies

A wide variety of methods for unification of data sources have been developed. Many examine different pieces of information available about the attribute. Others examine instance data. Most are separate methods that were not designed to work together. They are separate pieces and as such correlating the findings between methods requires an external, manual process. In order to use findings from one

method as an input to another method, the operator has to transfer the information between applications based on the assorted methods.

Unifiers of the Example-Driven Integration method are the beginning of automated data source unification. The method is rather limited in its abilities. The Example-Drive Integration method unifiers rely on human intervention to examine the data items. As stated previously, the method is too heavily reliant upon the human's ability and knowledge of the data sources. The major problem with this type of methodology is that the process is not automated to find new matches other than those easily identified through relationships in the schema. The unification process requires extensive amounts of time to find matches, an extensive knowledge of the data sources being unified, or both. The method may be used effectively on trivial data sources with few items to be unified. However, the very nature of this method limits its effectiveness on larger databases. The method will also be limited in effectiveness when used to unify several data sources or on data sources covering multiple, different domains.

Textual similarity methods attempt to compare specific words found in the attribute name. These methods are limited because they do not use available outside information nor any data population information. They will find matches,

but the findings will be limited to the attribute name. The textual similarity methods are quick to implement and are simple automated unifiers.

The Meta-data examination method took the next step in unification. The method uses the computer to examine things that are readily available about the data. It uses the information to make educated guesses about which items may be similar based on similar meta-data information. The method does not take into account the actual data. Another major disadvantage of methods using meta-data examination is that they do not allow for expansion of their capabilities [Clifton97].

The remaining methods are major improvements over those that use meta-data examination. These remaining methods use metrics that can be determined from the data contained in the data sources being unified and they make an attempt to give the data meaning or at least a meaning in relation to other items in the data sources being examined. Methods based on structure of the attribute, data items, tables or schemas and those that use graphs and structure matching take into account the structure of the data sources including proximity of the data items in the structure of the data source.

Although the structure of the attribute, data items, tables or schemas methods allow for expansion, they do not implicitly take into account known textual similarities found in the attribute name nor the description of the attribute item. They focus on the data items and try to learn information from the data items. If an implemented method cannot determine a match from these items, it may not find the match. This method may be good in identifying matches between attributes with cryptic names, but it derives no value from noncryptic names. In theory, it could miss a match between "phone_number" and "phone" if it is unable to find relevance in the actual data items that may be represented differently.

Graphs and structure matching methods are very dependent on the internal structure of the graphed schemata. If casual or implicit relationships are not defined in the original schema or are not imported properly, matches will be missed. These methods do not examine actual data information nor do they take into account any other pieces of information known about the attributes. They do not allow for expansion to take these items into consideration.

The linguistics method uses outside information to help expand the meanings of words or groups of words. Despite this strength, it does not allow for expansion nor does it use other methods to find additional matches. The method is

also limited to the information provided in the thesaurus. If the thesaurus is not complete enough to cover all instances of attribute naming, matches will be missed.

Combined approaches try to tie the previous methods together in a way that exploits everything that can be discovered about the data being examined. By examining both the data and the meta-data and combining the information derived, the method using a combined approach stands the best chance of finding meaningful relationships between the data from remote sources. The existing combined approaches either do not incorporate all existing methods, do not allow for expansion of the methods used to evaluate matches, or both. While these combined approaches are more advanced than the previous methods, they cannot be considered complete. For example, Cupid and MOMIS use different linguistic-based matching components and do not allow for expansion to use the component from the other method. If any match is missed by Cupid's linguistics matching approach but could be found by MOMIS's linguistic matching approach (or vice-versa), then the match is missed showing that the Cupid process is not complete. Table 3 shows specific deficiencies for the major unification methods.

Method	Deficiency
VIbE	Requires sample mappings; only reports exact matches; framework definition based on Java and FraQL
Delta	User has to hunt for matches based on responses to queries; uses textual similarities of attribute definitions; does not use instance data
SemInt	Studies found produced poor results on its own but was a good component when used with another unifier [Clifton97]; does not perform name-based nor structure matching; not designed to interface with XML documents
MOMIS	Not designed to be expanded to include other methods; design built on Corba
LSD	Only designed to work with XML documents; does not perform structure matching; not designed to be expanded to include other methods
SKAT	Not designed to work with databases; not designed to be expanded to include other methods
TanScm	Not designed to work with databases; does not perform instance level matching; can not be expanded to include other methods
ARTEMIS	Not designed to work with XML; does not perform instance level matching; can not be expanded to include other methods
Cupid	Does not perform instance level matching; not designed to be expanded to include other methods
Clio	Is a unification manager; not designed to be expanded to include other methods

Table 3: Existing Unification Methods and Major Shortcomings of Each Method

Additional short comings that were identified are:

1. Most of the examined methods do not allow the unifier to select which of the incorporated methods/features to use for the unification process.

2. All of the examined processes are rigid in the way they return their findings. The methods return only the answers they were programmed to find. Additional information is determined during the process but is not reported. Therefore, the additional information cannot be used in further analysis nor can it be examined for anomalies.
3. Methods, which incorporate multiple methods of evaluation, are rigid in the way they rank the findings.
4. The possibility of new, undeveloped data sources is not incorporated directly into many of the method definitions.
5. None of the methods offered an easy way to modify the abilities of the unifier.
6. None of the methods allowed for easy addition or removal of unification methods.

1.3 Statement of Problem

"Schema integration is the process of generating one homogenous database schema from several, heterogeneous source schemas" [Geist01]. As the number and size of remote data sources grow, it is becoming more difficult to combine the contents of these sources into a single, useful data source. There are many reasons to use remote data. However, data in one source may be duplicated in other

sources. Duplicate data items are not needed in the final unified schema. Only one source of like items is desired. To limit the number of duplicate items in the final schema, the duplicated items must first be identified.

Existing methodologies attempt to find possible duplications by examining the attributes and data items in different ways looking for similarities that are meaningful. Processes, which use one method of matching the information, can find simple matches but will miss other matches. These other matches may be revealed through the use of a different unification method [Doan00]. Processes that use more than one method will help to uncover more data matches [Madhavan01].

Data integration is not a process that can be fully automated [Miller01]. A human will be involved in setting up the sources to be compared, inputting any additional information needed during the comparison process, and evaluating the potential data matches [Madhavan01]. One of the goals is to lower the amount of time that a human must be involved to a negligible amount while exploiting the computer to find as many duplications as possible. One method or combination of methods will not be 'the best' method in all situations. A combination of methods helps to find additional matches [Madhavan01]. A match that can be detected by only one of five different methods may rank very

low in the ranked results depending on how the five method results are combined. This can be adjusted by making one method more heavily weighted but this adjustment may skew other results in the process.

The findings may also be skewed through the use of unneeded or unproductive unification methods. The raw data may be useful in finding these unproductive methods. Because the raw results are not available to the operator, neither casual observation nor statistical analysis can be performed to validate the need or usefulness of a unification method when used in a combined approach.

Some existing methods require specific operating environments or rely on specific languages. Some use experimental components that are not publicly available. Many only work with specific data source types. Currently, most methods of unification examine attributes on a 1:1 or 1:n basis at the attribute level. Some methods identify like tables or entire schemata. Other methods are very specific in the types of matches that can be found. As new methods of evaluating potential matches are devised, it is useful to be able to add these methods easily into the existing methodology framework. Many of the previously devised methods or combined methods do not allow for this type of extension.

1.4 Solution and Additions

dbUNiFier, the proposed solution, is a framework for the unification process. The framework is designed to allow for expansion of the types of data sources it can unify. It allows for both expansion and contraction of the methods used to perform the unification. Figure 1 shows how the data sources and unification methodologies fit into the dbUNiFier framework. The proposed method is platform and language independent. Most importantly, it treats the unifier as a tool and allows the operator a great deal of flexibility in how the unifier will be used as well as making both intermediate data and findings available for analysis.

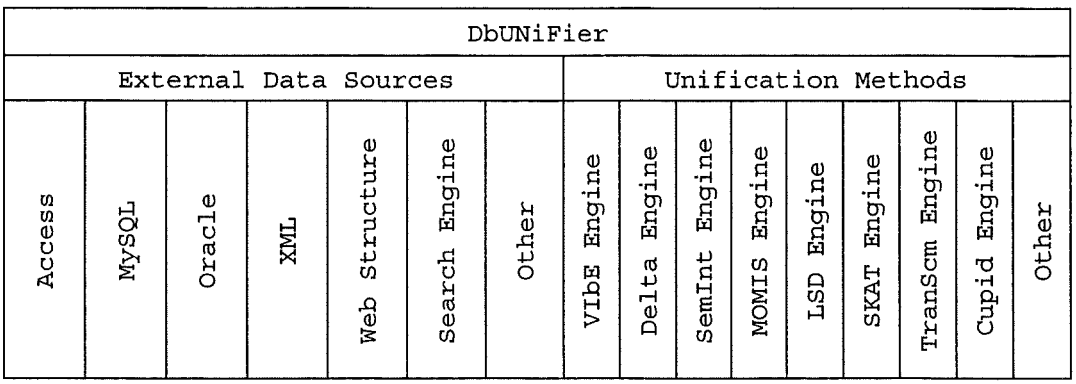


Figure 1: Existing and Future Data Source Types and Unification Methodologies Fit Inside the dbUNiFier Framework.

The design of the framework is built around the loose coupling principles of object-oriented design. Each major

component is a separate piece that can be modified independently of the rest of the program. By defining each major component as an object, the framework uses easily identified and expandable component parts. The interactive relationships and communication between these parts are defined below.

Data sources are treated like external objects. Communication with the data sources is handled by an insulating wrapper. The wrapper is used to translate the data found in the data source into a representation recognized by the unifier as a whole. By treating all data sources as an object and insulating them from the rest of the unifier application, it is possible to include a wide variety of databases, websites, documents, text files, Extensible Markup Language (XML) structures, or any other data source into this framework.

Because the goal is to create a framework, the platform for the unifier does not matter. The system environment will require enough resources to handle the unification being performed. The type of structure used, the language(s) used, the external communication methods, the operating system and other platform items are not defined. This allows the framework to be implemented in any suitable environment.

The framework allows for expansion of methods used but also gives the operator the ability to ignore methods once they are installed. Not all methods will be useful in all situations. By ignoring useless or trivial methods, the operator can save time and system resources. The goal of this feature is flexibility.

The possible loss of meaningful data is resolved by storing the data determined by the unification method(s). This data is made available to the operator. A side effect is the possible reuse of data among multiple methods.

The proposed unification framework is a methodology to easily allow cross platform implementation and development of unification methodologies. The framework defines a data access method, a data retrieval method, a presentation method, a reporting method, and a method of organizing individual unification methods so they will work within the defined framework. The framework definition in no way attempts to define a specific unification method. It instead provides a definition of how external unification methodologies can work together in an organized way and be automated as much as possible.

The goal of the framework is to allow for easy addition of new unification modules and metrics used by those modules. It defines the communication specifications between the

unification application and individual unification methods. These methods are added to the framework in a modular or object oriented way. The added unification method does not become a part of the framework. The added module provides additional functionality to unification software implementing this framework.

1.5 Components from Existing Methods

In an attempt to keep this unification framework as useful as is possible, it must be able to interface with a wide variety of data sources. Repeatedly in the examined methods, the ones that could unify the widest variety of sources used wrappers or similar interfaces. The wrapper concept is very important to allow the unifier to connect with existing and future data sources.

The user must be able to examine the results, possibly enter preprocessing information, and interact with the unifier to accept or reject potential matches. All methods use an interface for the operator to control the unifier. Different methods require different input from the operator so the user interface for the dbUNiFier method is used for all interactions needed from the user. This interface will be used to enter preprocessing data, select unification methods to use, select the known schema to unify, validate

matches, and examine the unification findings to determine additional matches.

This framework does not specify which unification methods will be implemented. It acts as a shell with the unification methods handled as functions of the framework. Since the framework cannot unify any data without these methods in place, any or all of the examined methods are essentially part of the completed unification framework design. The actual unification techniques from the previous methods remain as part of this framework. They become the component parts that actually do the unification. To work in this framework, they must be stripped of everything but their underlying unification function - the actual algorithm used to perform the evaluation. These methods accept input needed to complete their evaluation from the dbUNiFier. They report their findings back to the dbUNiFier. The dbUNiFier stores, combines, further evaluates, and presents the findings of each component.

The concept of an internal representation is in place but the actual representation is left to the implementer. There are a variety of reasons for this including the size of the sources being implemented, the amount of data found in those sources, the resources available, and the skill set of the implementer.

Actual mappings between the two sources are stored in a mapping table. The table stores enough information about the two attributes that can be used as a single attribute in the unified schema.

Chapter 2

The dbUNiFier Framework Methodology

The previously described unification methods and other methods like them are good ways to unify remote data sources. Generally, when each method was designed, the originator of the method had specific goals in mind or knew what type of data sources would be involved. Because of this, the methods were designed to fix specific problems.

Several of the methods took a wider focus for the approach. Their goal was to design a methodology that covered a wider variety of situations or were more thorough in their evaluation of potential duplications of data from different sources. The dbUNiFier methodology also attempts to be general in nature. It attempts to be so general in nature that it takes into account future unification and technology advances.

Every method examined has built upon previous methods to accomplish a task. Some were direct building blocks or steps. Others were more abstract and borrowed ideas from other unifiers or even from unrelated computer fields. Every one of them also had some fresh perspective on the overall situation of unifying remote heterogeneous databases. The dbUNiFier method takes into account the

findings of previous methods. The majority of dbUNiFier is bits and pieces of other methods. These are arranged in a more advanced, more comprehensive framework targeting the goal of a generic unification structure. The dbUNiFier allows for expandability as new methods of unification are devised and for the use of new technologies. It is more flexible and more robust than previous methods.

The field of automated heterogeneous database unification is evolving. Originally, the problem was simply unifying two databases, which is not without its complexities. Without solving that problem, it is nearly impossible to make it to the next phase in the evolution. Currently, the schema integration or unification field is in a phase that needs a way to unify a variety of data sources over different domains that will automate the task as much as possible and that will be as thorough as possible. Also needed is a higher level of flexibility, a modular structure that will allow for expansion of the abilities of the methodology without changing the individual methodology nor the framework structure, and a higher level of control to the unifier over how the unification will be performed.

The dbUNiFier application is constructed from seven major components. A Unification Method Handler controls the flow of the evaluation process and would be considered the main application. A Preprocessing module is used to collect any

items needed from the operator or other source prior to a unification attempt. The Data Source Wrappers communicate with external data sources. A Local Representation stores the remote data locally and acts as a repository for other important data. Any number of Evaluation Objects are used to perform the unification process. A Match Validator is used to allow the operator to validate potential matches. A Reports module reports the findings. If a GUI or command line interface is used, it will need to communicate with the Unification Handler and, if needed, the Preprocessing module, the Match Validator and the Reports Module. Figure 2 shows how the pieces fit into the dbUNiFier framework. Each sub-component interacts with the Unification Handler.

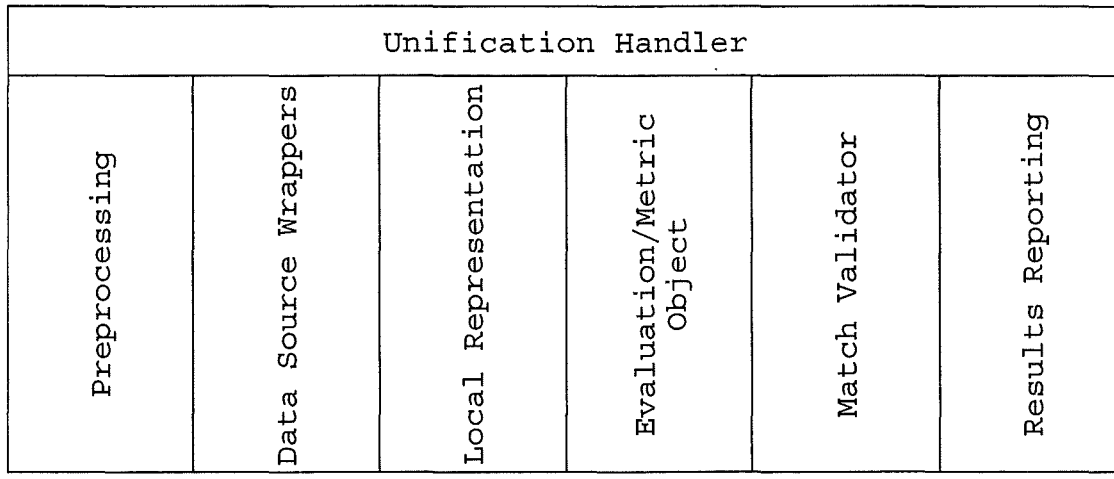


Figure 2: The Major Components of the dbUNiFier Application

The first step of the unifier process is to gather the information from the remote sources and make it local. To

perform the task at hand, the data source information and population must be gathered from the various sources. This should be gathered all at once so network access times will be eliminated in all but the single gathering of information from the remote sources. Since the framework is not concerned with the type of database or the access method, this portion is an object that can use any productive method to gather the information. When validating this framework, research found that database requests that selected all items in a single table are more efficient than requesting single items as needed (see the analysis in chapter 4).

The dbUNiFier acts as a batch system with a definite beginning and a definite ending. The information used to determine matches is gathered at the beginning of the run. This information may not be valid after the batch is run if one or more of the data sources are modified after the unification evaluation is complete. Live data sources may update with new data and possibly new attributes or tables. When this happens and a new unification evaluation is needed, archived data sources should not be reexamined at the data source level. The results of a previous unification will be available in the local repository.

The information is to be stored locally in objects. Again, the framework is not concerned with the representation of the data as long as the data is textual, it is stored, and

it is easily retrieved. A handling object that insulates the local storage from the unification methods can then request pieces of information. The handler controls the flow of the unification process.

Metrics modules examine the data looking for information needed by the dbUNiFier to be used during the unification. If a specific metric is not needed (selected), then it will not be evaluated. The results of the analysis are stored for additional use during a unification attempt and for future use or analysis. The dbUNiFier loads the needed data into a metric object and executes the metric. Once the calculations or comparisons are performed, dbUNiFier gathers the analysis results. The results are passed to the local representation where they are stored. Since the process is object oriented, additional metrics modules can be included easily as long as they conform to the proposed object architecture. Additional tools, such as new thesauri or dictionaries, can be interchanged or treated as a new module when determining metrics used by individual unification modules.

The user is able to specify which, or all, of the methods to utilize when unifying the data sources. This enables the user to run all the unification modules to fully examine the sources for duplicates. This also allows for modules that cannot handle the specific type of data in an effective way

to be turned off. For example, if the databases contain only numeric data entries, it is not necessary to invoke a module that is known to only work with textual entries. This will save time in running the application and will avoid introducing invalid findings.

When using metrics that require multiple iterations of the evaluation process, the user must be able to iteratively accept or reject a potential match. The Match Validator allows the user to do this. The validator pulls all results, combines, and weights the scores based on predetermined weightings, and presents ones that are over a certainty threshold. The weighting of each metric piece should be modifiable by the user.

The information discovered by the process is available to be presented to the unifier in its entirety. The finding for a specific pair of attributes is assigned a certainty value by each of the modules enacted. A certainty value is then given to the two attributes that takes into account the findings of each individual module as defined by the metric definition. Each piece of information can be returned to the user for evaluating potential matches. This will enable the user to examine the data at face value to find matches and will also make the results available to the user to run external evaluation applications on the findings. The

operator is able to examine the findings of individual metrics.

When a match is validated, the match is added to the local representation mapping table. When a match is discarded as being incorrect, this is stored in the match validator. Additional iteration results are compared against the mapping table and the discarded table. Only unresolved potential matches are presented to the user to validate. If a potential match has been previously resolved by the user, it is skipped. On second and successive iterations, only metrics modules that can provide additional matches will be invoked. If new matches in the mapping table can not possibly alter the metric modules findings, it should not be run again. This must be determined on a per module basis when the unifier is being built.

A series of objects designed to produce reports about the unification attempt can be used to present pre-designed reports at the end of the unification process. If the dbUNiFier is utilized as a true batch application, these reports will be predetermined in the coding. If the dbUNiFier is implemented with a GUI, the operator should be able to select which reports should be generated. Since the report generation is handled by an object, the output of the report will be programmed into the object. All information needed for the report must be collected from the local

representation or determined by processing portions of that information. The output method will be dependent upon the implementation. The reports can be stored, delivered, or presented in any manner (such as a report posted to intranet page, Internet page, email list, in Excel format, etc).

The key to the success of this methodology is in the object-oriented approach. Additional unification modules can be added or removed as long as the modules adhere to the defined structure and use specific methods to pass information back and forth. Once the information is passed to the module, then the module can use the information in any way that is needed. After any evaluation has been completed, the results can be retrieved to the main program by using other defined methods.

2.1 Unification Handler Definitions

The Unification Handler (UH) acts as the controlling mechanism. It is responsible for the flow within the unification application. This object calls the Wrappers and begins the localization of the remote data. Once the wrapper has collected the information and data, the Unifier calls the local representation to load the data into the local data store. The Unification Handler is also responsible for the flow of which installed unification methods are used on the data. Once the selected unification

algorithms have been applied to the data and structures, the Unifier activates the Match Validator. Depending on the user's input, it runs the unification method or methods again taking into account the user's inputs (if the method being used allows for this). Once the user is satisfied with the results, the Unifier calls the Reports Modules.

If an interactive design is required, a GUI or command line interface communicates with the Unifier module. The interaction allows the user to control the flow of the program. Otherwise the design is that of a batch application that follows the steps outlined above without allowing the user the option of validating the matches. In this case, the matches are determined externally by examining the reports. Matches found in this manner are input manually into an external application or any mapping table. The Unification Handler is responsible for all communication between the wrappers, internal representation, the unification metrics, the match validator, and the reports module. The UH must have pathways defined that allow the appropriate data to flow freely between these objects as needed. There should be pathways between the unifiers and the local representation; the local representation and each metric; the local representation and the match validator; and the local representation and the reports modules. If a GUI is being used, the UH must have the logic to exclude modules that are not invoked by the

user. Otherwise, the UH will have the modules to be used hard coded into the implementation.

The dbUNiFier follows a discrete series of steps. These steps are sequenced by the UH. A preprocessing step allows the user to specify target data sources, to input any information needed by metrics objects, to pick from included metrics to execute during the unification process, and to generate any reports upon completion of the unification run. An evaluation step then executes the selected metrics. Results from the evaluation are presented to the user through the Match Validator. Iterations of the evaluation-validation steps are repeated until the user is satisfied. Then, the reports are generated.

2.2 Preprocessing

Many of the examined unification methods require information input by the user prior to the beginning of the unification process. This includes known mappings, data dictionaries, thesauri, or other pieces needed for the method to perform. The collection of this information is abstracted into a separate object class. The preprocessing object can be broken into any number of cohesive smaller objects as is practical. Since different methods require different processing tasks, it is not possible to define the individual preprocessing tasks as a part of the dbUNiFier

framework. Instead a placeholder is provided to attach the needed functionality. Information collected by the preprocessing step must be stored in the local repository.

2.3 Data Source Wrappers Definitions

Wrappers are communication ports between the unification application and any data source. Each type of data source will have a different wrapper. The wrapper is used to connect to a data source, to request any data from the source, to convert the data into a common format, and to report the data back to the unification application. Wrappers are designed and written for a specific data source type. While assorted data sources may need to be accessed differently, every wrapper will return data in the same manner. The wrapper functions as a data gatherer and translator.

The wrappers need to create an access connection to the data source. Data source connections will vary among source types and among languages used to implement this methodology. The implementer will need to research for information on connecting to a specific data source type and the implementation language being used. A wrapper for a networked database, such as MySQL, Access, or Oracle, is a straight forward piece that opens the communication, then requests the structure and contents of the database. The

wrapper object will be instantiated and it will communicate directly with the remote database application. The wrapper uses a database communication package such as DBI for Perl or ODBC for Java. Figure 3 shows how the wrapper lies between the data source (Oracle Server) and the unifier application. The wrapper communicates with both the data source and the main unification program.

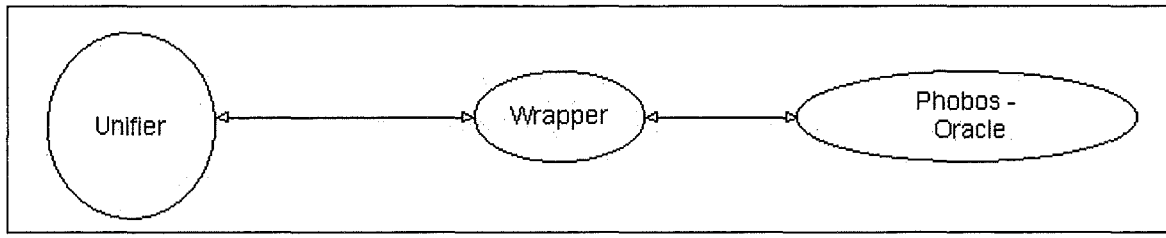


Figure 3: A Wrapper Insulates the Unifier from the Database

Wrappers for textual data coming from non-structured or semi-structured data sources will need to pull the desired contents of the data source. Files in a HTTP environment can be simply read. This includes any flat file, website, XML document, Peer-to-Peer network, or any other retrievable piece of data or information.

The goal with the use of wrappers is to simplify and standardize the internal code and structure of the main unification application. By separating the main application from the data source types, a reusable object will be manufactured. Once a wrapper for a common data source type

has been built, it can be used to easily connect the application to any other data source of that type. The wrapper must be able to communicate with both the data source and the unifier application. At some point between the grabbing of the data from the remote source and reporting the data back to the main application, the data must be converted to the common format used by the main application. For example, dates may need to be reorganized or converted from seconds to year/month/day format. Data types may need to be converted if one data source refers to the data type differently. Data type equivalencies will also need to be mapped. Other conversions may be necessary. If any security or encoding scheme is needed to communicate with the data source, the process is performed by the wrapper when it is not handled by external processes between the unifier and the remote data source.

The data source wrapper is responsible for determining any meta-data needed for any unification metric module. If the meta-data is readily available, it is collected. If not, it is determined. For example, the number of instances stored in a table. If the data source does not provide this number, the wrapper must implement the necessary processes to determine the number. The goal is to get the data information to the client machine. Any method, including writing the information to a file and then using FTP to fetch the file, is acceptable as long as the information can

be extracted from the data source, converted to the local storage format, and then loaded into the local storage.

2.4 Evaluation/Metric Object Definitions

The unification process is a series of comparisons or functions applied to the data found within or among data sources. To evaluate these comparisons, any given unification method uses at least one metric. The metric signifies either equality, a percent chance of a match, or some other piece of data to be applied to a subsequent evaluation. The metrics used by dbUNiFier can be the schema matching algorithm components from previously defined unification techniques, new pieces designed for specific environments, or any newly defined methods. Any evaluation piece can be used as long as the method requires the data or information from the data sources and then, evaluates the data based on some formulation and metric.

The UH controls the flow of communication between the main program and the evaluation pieces. The piece must be able to receive the required data to perform the evaluation. Upon completion of the evaluation, the object returns a collection of the items used in each portion of the evaluation and the result of the evaluation. For example, a comparison of every attribute in one data source to every attribute in another source returns an entry for every

comparison and the result of the comparison. Each evaluation method is implemented as a separate class. If two evaluation techniques use similar components, these components should be sorted out as a separate class. The common component is then called by each evaluation piece when needed.

An assumption has been made that the unification environment will have the resources needed to perform the unification processes. If the unification is to be conducted between large databases, it may be necessary to only keep abbreviated or summarized data regarding the evaluation. In extreme cases, this feature may be disabled for some or all of the evaluation classes. If this feature is disabled, some of the functionality of the dbUNiFier is lost.

2.5 Local Representation Definitions

The Local Representation (LR), or local store, is a repository of all information and data needed to complete the unification process and the results of the unification process and sub-processes. The main collection in the local representation is an equivalent copy of the data and information found in each of the remote data sources. It also includes information about the source of each piece of data, results from comparisons and calculations, and the mappings that are generated by the unification process.

The logical representation method for a generic unifier is the use of a local database. As data is transferred from the data sources, the data structures are rebuilt locally and the data is populated into the local data store. Storage in local memory is superior in speed but local memory is finite so a local representation built around local memory storage will not be as scalable as one based on a database. While a memory based representation can be used, the database method is preferred. The local representation of the data from the remote data source must be adequate to meet the needs of the evaluation methods implemented. Any features needed by the evaluation method, which are not present in the data storage method, must be emulated. For example, if the local representation cannot provide information about the size of an attribute, the software being implemented must be able to determine this information.

Once the data is formatted by a wrapper, it is loaded into the local repository through the UH. The data can be stored in any logical manner that keeps each set of data from different data sources logically separated. Each attribute and the meta-data for each attribute should be stored in the database if this information is needed for any of the unification methods to be applied to the data sets. All data from a single data source should be loaded locally in a

serial process. The data should be transferred and loaded as parallel processes or as a single serial process. The goal is to save communication and transfer time by collecting all remote data as a single step of the unification procedure.

Additional storage will be needed for storage of final metrics to be reported to the user. This information should be stored in a logically separate data store. The information stored should include the data sources, the tables, and the attributes involved. Since some metrics make use of instance data, it is up to the implementer of the framework to determine if this extensive amount of comparison information should be stored and presented to the user as needed.

Information used to determine metrics should be stored in local memory and released when the evaluation is completed. Information to be reported to the user in any way should be stored separately in an easily accessible structure (such as, an array).

The final mapping table must include the source of the attributes, information about the location of an attribute within that source, and the attributes themselves. In instances where the source and the source structure are the same, this information should be duplicated or flagged.

Information collected as part of the preprocessing step should be stored in the local representation. Pieces of information that can be used in future unifications should be allowed to remain in the repository from run to run.

2.6 Match Validator

The match validator (MV) presents the automated unification findings to the operator. The user approves or rejects potential matches. The approved matches are used as input in subsequent iterations of the unification analysis. This piece allows for any logical presentation of the information to the user and it allows for any logical input of the user's decision on the match. Upon completion of this step, the user should be able to continue the analysis or end the analysis and complete the post-processing tasks.

Communication between the UH and the MV must be available for all process results to be transferred to the MV. The MV will then pre-evaluate the data to present the user with a combination of attributes, the weighted or raw metric result, and the option of accepting or rejecting the combination as a match. An algorithm defined by the implementer of the framework will take all unification method metrics into account and assign predefined weights to those metrics. The pre-evaluation weighting and combination should be flexible and allow the user to define the data

pieces used in the pre-evaluation and assign weights to those pieces. The user should also be able to look at raw metric scores from each of the unification methods and possibly uncover hidden matches.

Once a combination of attributes is accepted, that combination should be added to the mapping table found in the local representation. If a combination is rejected, it should be stored in the MV. The next time the MV is presented to the user, only new potential matches should be presented. By keeping track of previously rejected matches, the user will not be bothered with re-examining the rejected items.

2.7 Results Reporting

Reports of the findings of the unification process can easily be provided to the user's organization. The information found in the local repository will be the context for the reports. A report could consist of any subset of the information or statistics on the process. A report of the uncovered mappings can be generated from the mapping table. A full examination of the contents of the local representation can be provided. Details of the evaluation metrics can be generated to help find errors in the algorithms. Specific reports are external to the dbUNiFier framework. Instead, a placeholder is provided to

enable the implementer to define their own reports. Requirements will vary upon the type of report requested and the delivery of that report. Reports can be customized for each situation so a library of report types can be generated by creating separate classes for each report.

Chapter 3

Implementation Notes

In an attempt to demonstrate key components of dbUNiFier and have the ability to analyze them in a working environment, a model was built to specifications found in this document. The source code for the application can be found in Appendix A. The software is not a complete implementation of a final version of dbUNiFier but is complete enough to perform the automated unification process and find potential matches. Appendix B contains the output from a sample run of the dbUNiFier implementation. Additional software pieces were written to examine key components of the dbUNiFier and compare them to the existing unification framework MOMIS.

3.1 Overview (software/hardware)

dbUNiFier analysis software was written entirely in Perl and uses MySQL for a local database. It was executed on a Pentium III computer with Windows 98 as the operating system and 256 MBytes of RAM. All nonessential operating system processes were terminated. The software was written in an object-oriented fashion with separate classes for each of the main modules. Measurements were taken while running in a MS-Dos prompt window.

An emulation of portions of MOMIS was also written since the framework portion of the MOMIS method was being evaluated, a full implementation of MOMIS was not needed but pieces of its architecture were required. All human interaction and data access portions were stripped. The actual unification portions were replaced with simple metrics modules.

dbUNiFier acts as a framework that can use the underlying MOMIS unification algorithms. DbUNiFier can also use a variety of other unification algorithms in the form of separate classes. Because these classes can be added or removed, the examination is not of the unification piece itself. The analysis is of the framework that houses the unification algorithms in both MOMIS and dbUNiFier.

The applications were designed to include all important components of the unification process that are affected directly by the dbUNiFier framework. Since preprocessing is dependent on the modules being used and this portion primarily requires the use of human time, it was not examined. Preprocessing is simply the entering of data.

An exact duplication of the Wrapper modules was used in the testing software for both the dbUNiFier and the emulation of MOMIS. Both use the same local representation method with the understanding that different representation methods

could have been used as long as the same method was used in both pieces of software.

Both analysis applications used the exact same metric classes. These classes are simple algorithms like the ones used in many unifiers. The understanding being that since both applications were using the same match evaluation techniques then comparisons of the time data would show significant differences in the structure. Neither test unifier was designed to actually complete the unification process. Both were designed to aid in the examination of how the pieces interacting with each other cause time variations between the two frameworks being examined.

Neither test application performed the match validation process. Again, this process is more human interaction than it is automated computing. It is assumed that two operators would use the same amount of time to examine the same set of match data from the same set of metric evaluation classes.

MOMIS does not include a reports module so this feature of dbUNiFier was not examined for processing and generation time. The reports module was not included in the dbUNiFier.

Test applications were run in the graduate laboratory at the University of North Florida. Tests used two remote data sources. The first source was Marketingbots.com which is an

implementation of MySQL running on a server based on the Linux operating system, a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world [Webmaster03]. The server was accessed via the Internet. The second source utilized was Phobos.cocse.unf.edu which is an Oracle implementation running on a separate Windows NT based server. It was accessed through a local network. Information on network access time can be found in Appendix C.

3.2 Unification Handler

The main class of the implementation of dbUNiFier is the Unification Handler (dbUNiFier.pl). The class first specifies which other classes will be used as part of the application. These classes are wrappers::marketingbots; wrappers::phobos; representation; methods::attributename; methods::attributenamesubstring; and methods::attributeinstance. As with all Perl applications, the class name signifies the location of the class file and name of the class.

A diagnostic file is opened to help keep track of the program path and data being accessed. This diagnostic file was used during development of the application. It provides no added value to the program, other than a simple way to find errors.

First, the dbUNiFier retrieves the contents of the remote data stores. Separate objects are instantiated for each of the wrappers needed. A pointer to the location of the object is stored. The object for the local representation is also created and the location of the object stored.

The content from the remote data sources is loaded serially. All data is collected from one source before the contents of the second source is collected. The UH calls the wrapper objects and the data received is then sent to the local representation object where the data and information is stored.

The unification metrics are called serially. All data needed to determine a unification metric are requested from the LR and loaded into the metrics object prior to the evaluation. This loading process is handled by the UH. Once the metrics are determined, the UH receives the results in an array and forwards the array to the LR. The first two metrics used require the same data to perform the analysis and return similar result structures. It was possible to use the same function to load and unload both objects. The third metric required more data from the data store so a separate function was written to handle the data loading and retrieval of the results. After one metric is completed, the next is loaded and the evaluation performed. Upon completion of the unification attempt, a report is generated

containing the contents of the repository. This portion of the run is not included in the time measurements.

3.3 Data Source Wrapper

In order to communicate with a database in Perl, the DBI module must be installed. Additionally, communication in this implementation is with both Oracle and MySQL servers so the modules DBD::Oracle and DBD::MySQL were also required. Connection through the DBI module consists of identifying the data source name or location, the user name, the password, and if needed, the database name. Connection with the Oracle host is made directly from the client application. The database connection is made directly from the client machine.

Due to security features in place, the Marketingbots.com MySQL server cannot be accessed remotely. The wrapper used for this data source is a distributed process with a portion running on the client machine and a separate portion running directly on the data source machine. Figure 4 shows the structure of the wrapper. The client machine communicates with the server through HTTP based communication between the wrapper pieces. The actual database access is made from the portion of the application running on the server machine. The requests are sent and the required information is ported back to the client via a HTTP Pathway.

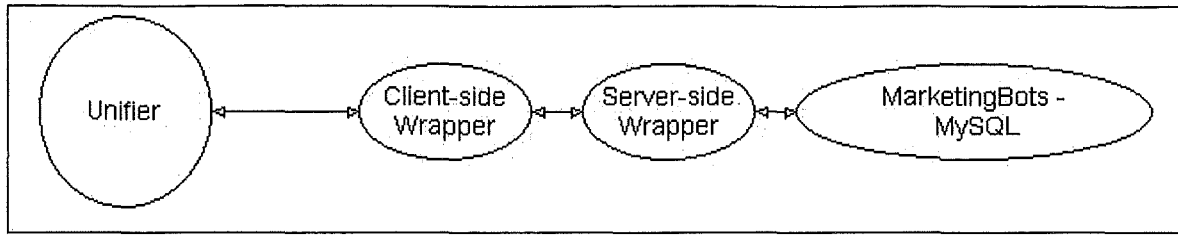


Figure 4: Distributed Wrapper Communicates Internally Across the Network

3.3.1 Oracle Wrapper Communication

Connection with Phobos is established by setting the server location, user name, and password. The connection string is assigned to a file handler ($dbh). All communication with the remote server is then referenced through the file handler.

```
$dbh = DBI->connect("dbi:Oracle:host=$host;sid=ORCL", $user, $passwd);
```

Commands (queries) are sent by loading the SQL statement into a statement handler and then, executing the statement. Errors are caught when the statement is executed.

```
$sql = "select * from profile";
$sth = $dbh->prepare($sql);
$sth->execute or die print "No info in database \n($DBI::errstr)<br>";
```

In a case where multiple results are being returned, the client application simply iterates through a request function performed on the statement handler. Each iteration advances the pointer to the next tuple being returned.

```
while (my $ref = $sth->fetchrow_arrayref()) {
        print "Found a row: id = $ref->[0], name = $ref->[1]\n";
    }
```

In instances where a count is being performed or any other single item will result, the request function is called once.

```
my $ref = $sth->fetchrow_arrayref();
print "Found a row: id = $ref->[0], name = $ref->[1]\n";
```

Appendix D shows conversions needed when transferring contents of an Oracle database to a MySQL database.

3.3.2 MySQL Wrapper Communication

Connection to the MySQL server is a similar process. Perl uses a single Database Interface (DBI) for access to major databases. Additional functional requirements are specified in separate modules that are implemented for the specific database type. In this implementation, the MySQL wrapper is

distributed between the client and server machines. A signal is sent to the server portion of the wrapper by the client portion of the wrapper. Commands sent are predefined (connect, query, disconnect). The server side portion receives the call via HTTP request with the command included in the call (such as, http://server/wrapper.cgi?command=1). The server side performs the required function. It returns the retrieved data or message to the server as a text string. The client side portion of the wrapper decodes or reformats the data into the local representation format.

Connection to the database is located on the server side portion of the wrapper. Because it is using a database that is local (to the software portion), no host name is needed.

```
$dbh = DBI->connect("DBI:mysql:$database","$user","$passwd")or die print
"$Mysql::db_errstr<br><br>";
```

As with the Oracle wrapper, the query is prepared and executed.

```
$sql = "select * from EMAILS limit 100";
$sth = $dbh->prepare($sql);
$sth->execute or die print "Query failed n($DBI::errstr)<br><br>";
```

Multiple responses are also received from the database in a similar fashion.

```
while ( @columns = $sth->fetchrow) {
        print "$columns[0]|$columns[1]|$columns[2]\n";
    }
```

Since this request was made from the client side, the print statement in the previous code block will print to HTTP as the default I/O medium. A single item was received in the same manner.

```
    @columns = $sth->fetchrow;
```

3.4 Metric Evaluation Objects

For this analysis, there were three simple metrics built under the dbUNiFier framework. These are not the advanced metrics used by the advanced unifiers. These are simple metrics that that will uncover some simple match types. They are components that are in place to help simulate the unification process and produce results that can be examined to show how the unifier works. The first metric is a comparison between the attribute names to see if the attributes have the same name. The result of the comparison

is a 1 if they are a match. The result is a 0 if they are not.

The object has memory structures for all the attributes from one source and a second set of memory structures for the second source. Each set is represented in three arrays with position 0 of each array referring to the same attribute. A source array keeps track of the data source from which the attribute originated, a table array for the table location of the original attribute, and an array for the name of the attribute. Any memory structure would have worked to keep track of the source of the attribute. Multiple arrays were used because of the ease of traversal. The same structures are used in both analysis pieces of software. A third set of arrays is created to store the results of the metric evaluation. FINAL1 and FINAL2 store the two attributes being compared. RESULT stores the result of the metric evaluation.

The memory structures are loaded from the UH which retrieved the data from the LR. The metric loops through the arrays and performs the evaluation. The results are stored in the three arrays. Position one of each array refers to the items in the single evaluation. Upon completion of the analysis, the UH pulls the results and sends them to the LR. The second metric object is loaded in the same manner as the first metric. The object evaluates one attribute name as a

sub-string in the second attribute name and vice versa. The results are stored as in the first metric. The results are then transferred through the UH to the LR.

The third metric examines instance data. It is loaded with one set of instance data per data source. A comparison is made between individual attribute items. Then, the next set of data is loaded. A single memory structure keeps track of metric scores. After the first comparison is made the contents of the next attribute of the second data source is loaded and the comparisons done. The full set of scores from the process are retrieved by the UH and fed to the LR.

Originally the comparisons within the metrics modules were made only if there were equal numbers of instance items for each instance item. The metric would have found duplication datasets. The check for equality in size was removed to allow the program to generate larger amounts of internal data. A counter is increased when a match is found on the instance level. The total matches are divided by the number of comparisons made to determine a percentage of equality. The results are loaded into a local memory location.

In each metric class, a method *new* generates the structures needed for the analysis. The data is loaded directly into the metrics object from the UH. A class called *analyze* invokes the metric algorithm that examines the data.

3.5 Local Representation

Local representation for this implementation is using MySQL on a Microsoft Windows based computer. A local copy of MySQL was installed and configured. Communication from the unification application to the database is through Perl module DBI::mysqlPP. Connection and interaction with the local copy of MySQL are the same as those for the Marketingbots.com wrapper except there is no need to distribute the local object. Connection with the database is established when the object is created. The connection remains until the termination of the program.

Memory is set up for every item coming in or going out. Some structures are reused for both an in and out process. Specific arrays are ATTRIBS, TABLES, SOURCES, ATTRIBUTES, FINAL1, FINAL2, RESULT, INSTANCES, and INSTANCES2. Single memory locations are set up for the BOT reference (used for the communication with the local database), ATTRIB, ATTRIB2, SOURCE, SOURCE2, TABLE, and TABLE2.

New sets up the communication with the database and all structures needed to run the local representation. Create sets up the table used to keep track of all external attributes. This table needs to be rebuilt each time the program is run. It first attempts to delete the table if it exists from a previous run. Then creates a table to keep

track of the data source of the attribute, the table source, the attribute name, and the definition of the attribute including size, null, and default value. Addats controls the flow of the processes needed to add attributes into the local repository as attributes in tables and as entries in the meta-data repository. Checkifitexists checks to see if a table already exists when an attribute is to be added and makenewtable is used to create the table so the attribute can be added. Addattribute adds individual attributes to their respective table. Indexattribute adds the attribute and the meta-data about that attribute into the meta-data repository. Addinstances adds the instance data to the repository in their respective tables. Getallattributes pulls all the attributes and loads them into a memory structure so they can be accessed by the UH. Getallsources returns all the sources in the local representation and getalltables returns all the tables. Storeresults builds a tables for the results of a specific metric object and stores those results for future use. Getinstances retrieves all instance data for a specific table in the first source in the current unification attempt. Getinstances2 retrieves all the instances from a table originally from the second source in the current unification attempt. Cleantables is a house keeping function to remove a fake attribute defined when creating a table in the repository. When a table is created in the MySQL database at least one attribute is required. For simplicity in the design of the analysis

software, a fake attribute is added and then removed after the repository is built and loaded.

3.6 Results Reporting

The results object creates any reports that are required or desired from the analysis and unification process. In these tests, the reports were not generated. In real world use of the dbUNiFier, these reports would be needed. For analysis of the MOMIS and dbUNiFier frameworks, a reports module was not needed.

Chapter 4

Pre-Analysis

Prior to implementing the dbUNiFier a pre-analysis was completed to examine the relationship between types of connections to the database and how the data is requested. A series of tests were performed. Each was designed to examine a connection method and a request type. The tests were conducted on a Pentium III computer with Windows 98 as the operating system and 256 MBytes of RAM.

4.1 Description

When working with a database, a connection is made, the data is requested, and ultimately the connection is terminated. For this experiment, two schemes were examined. The first scheme opened a single connection and requested the needed data. All data handling was performed and then the connection was terminated. The second scheme opened a connection every time data was needed. The data pieces needed were requested and the connection was dropped. Subsequent data requests required an additional connection. In the single connect scheme, any additional data could just be requested because the communication channel remained open.

The second area of examination was in the method used to request the data. A series of tests were conducted by requesting each single item from the database and a second series of tests requesting all the required data in a table at a single time. The first tests queried the data with a query like 'select * from table where attribute=variable', stored the response in a series of variables, performed the comparison, and then, requested more data. The second set requested the data with a query like 'select * from table', stored the response in an array, and performed the comparison tests.

To simulate the automated unification process, a simple comparison of the two data items was determined between every pair of data in the dataset. A simple comparison of the two items represents a more complicated component. The assumption is that the unification method component will take x amount of time under any framework as long as the data needed is present. Therefore, the analysis is on the component needed to implement the dbUNiFier framework.

Time spent on the operation was measured within the software. A timestamp was made before the task being examined and a second timestamp was made after the completion of the task. The difference between the two timestamps was considered the time required to perform the task. A small amount of time is consumed in making the

calculations but since this time consumption is present in all tasks, it does not have a bearing on the results. The use of a database for the internal representation and the external data sources was predetermined. The goal of this analysis was to determine the best way to handle the data being gathered and used in the unification process. Source code for the pre-analysis software can be found in Appendix E.

A data set of 20,000 pairs of items was built for this analysis. Experiments were run with sub-sets of 500, 1000, 2000, 3000, 4000, 5000, 10000, 15000, and 20000 pairs to be compared. The data for the run was loaded into a database. The data was then extracted from the database into local memory, the comparison of the two items conducted, and the result stored in memory. A sample of the test data used can be found in Appendix F.

Each test was conducted 50 times. The time measurements were averaged over the tests of same type to get a normalized measurement for the particular measurement being taken. The entire series of tests was conducted on each data subset. Then, a second iteration of tests was performed. For information purposes, two other measurements were calculated. One measurement was the time needed to compare the two pieces of data if the data is already in

local memory was determined. The time to load the data into the database was the other measurement.

4.2 Results of Pre-Analysis

It was determined early in the analysis that the largest time measurements were being found when multiple connects were needed to access the data in the database and each data set (tuple) was requested separately. The time to conduct these two tests consumed over half the total time. Only twenty runs of these two items were conducted. The average on the two cover only those twenty measurements.

Table 4 shows the results of the analysis. The first column shows the number of items in the data set being used. The second column is the measurement of time to load the data set into the database. The third is the time to do the comparison if no database is involved and the items are already in memory. The detailed results for this analysis can be found in Appendix G.

Columns four through seven are the measurements of the tasks when a connection to the database already exists and is left open during the process. Columns eight through eleven are the same tasks as four through seven, but this time the connection does not exist prior to the first measurement. The measurement is taken, the connection established, the

tasks performed, the connection is closed, and a second measurement is determined. The time measurement represented in columns eight through eleven include the connection/termination overhead. Columns four and eight represent the time it took to request each tuple individually and perform the comparison. Columns five and nine are the time required to request all tuples from the dataset in a single SQL statement. Columns six and ten show the time required to select each item individually but not perform the comparison (assuming a comparison has already been calculated and the result is already present in the local repository). Columns seven and eleven are the time it took to request the dataset in a single SQL statement and not perform the comparison (for the same assumption).

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number of data sets	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
500	1.084	0.002	1.760	0.166	1.749	0.159	7.747	0.176	7.738	0.168
1000	2.167	0.006	3.697	0.331	3.674	0.321	15.685	0.336	15.673	0.328
1500	3.246	0.008	5.673	0.503	5.640	0.474	23.655	0.510	23.647	0.482
2000	4.328	0.012	7.938	0.669	7.903	0.633	31.951	0.666	31.916	0.647
2500	5.408	0.013	10.424	0.841	10.38	0.789	40.458	0.833	40.386	0.811
3000	6.482	0.027	12.773	0.997	12.718	0.946	48.81	1.013	48.737	0.957
4000	8.641	0.028	18.200	1.330	18.116	1.277	66.259	1.343	66.198	1.281
5000	10.804	0.04	24.868	1.672	24.774	1.59	84.851	1.666	84.935	1.588
10000	21.623	0.061	53.835	3.324	53.593	3.183	173.431	3.309	173.233	3.185
15000	32.427	0.091	80.762	4.98	80.393	4.766	260.399	4.922	259.720	4.739
20000	43.221	0.122	107.627	6.621	107.068	6.337	346.797	6.552	346.295	6.307

Table 4: Time Required to Perform the Examined Database Access and Comparison Tasks

When using a single connection, it was found that selecting all data as a single statement increased the efficiency of the application considerably. Figure 5 shows that both tests involving a single connection and requesting all the data from a table at once remained under ten seconds in the tests with the largest data sets. The set of results when requesting each item separately required more than ten fold the amount of time.

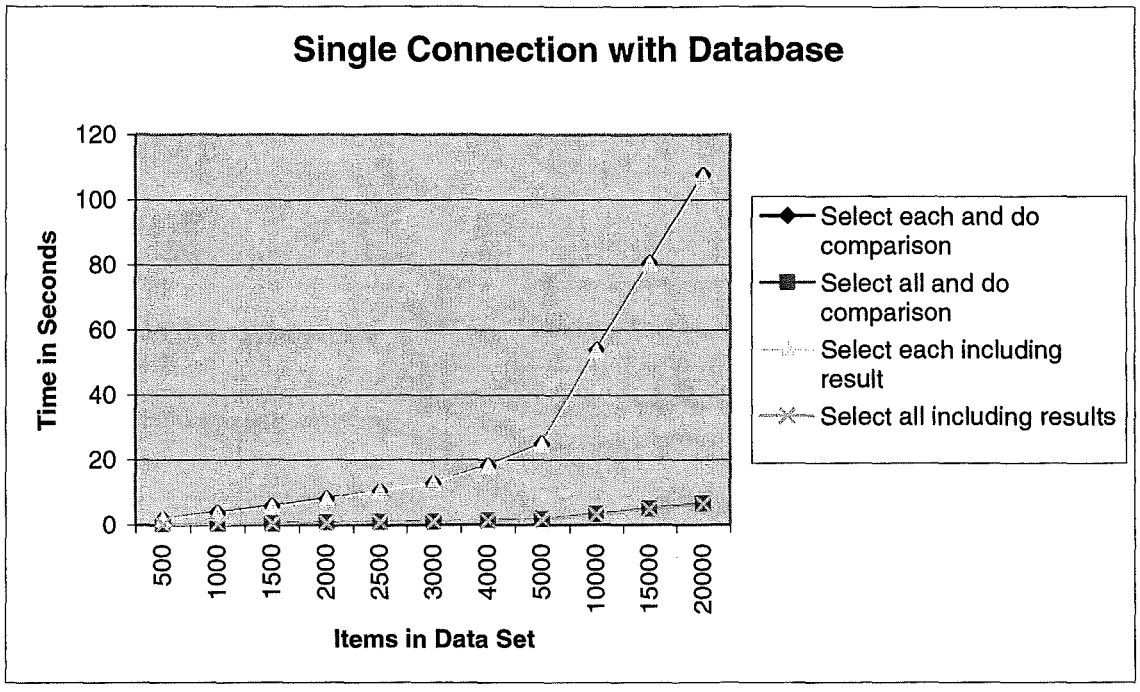


Figure 5: Time Required to Request Data from a Database when a Single Connection is Used

Multiple connections with the database had similar results. Requests for individual data items took considerably longer than obtaining all the data in a single request. Figure 6 shows that both methods requesting the data items individually were clustered and took much longer than the other two methods. The lines with the sharpest curve represent the requests for single data items. The clustered lines with the gradual curve represent the methods requesting the data in a single request. By examining the two graphs, it is obvious that when requesting data from a database, it is best when time is the metric to request the data in a single statement. The results of the tests conducted, for data sets of the sizes used, show that a single request is more efficient.

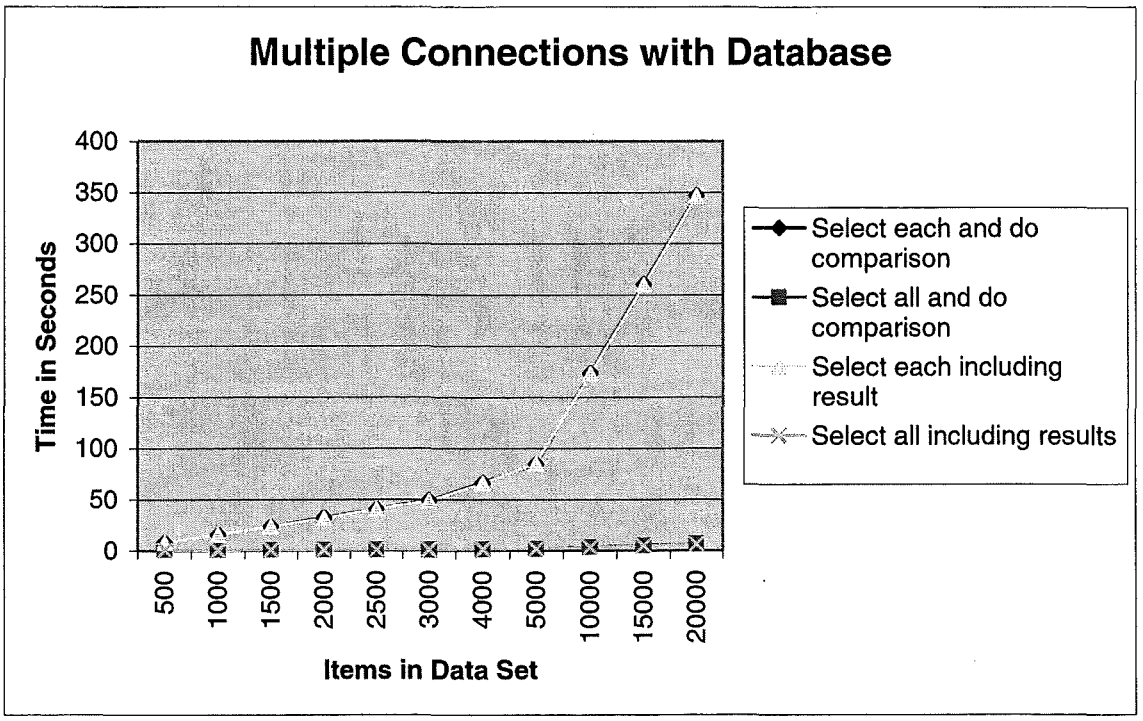


Figure 6: Time Required to Request Data from a Database when Multiple Connections are Used

Figure 7 shows the four preferred methods of handling communication with the database. These findings affected the design of the dbUNiFier framework and are found in the wrapper definition and the local repository definition. The wrappers collect all the information about a database in one short session, instead of requesting the pieces of information and data as they are needed. The local repository includes a single table devoted to location and definition meta-data of each attribute. This enables a single request to pull all attributes, all attributes from a specific table, attributes with specific definitions, or

other collections that are used in the unification process. Each of these methods takes about six and a half seconds to obtain the data, load it into memory, and make the comparisons.

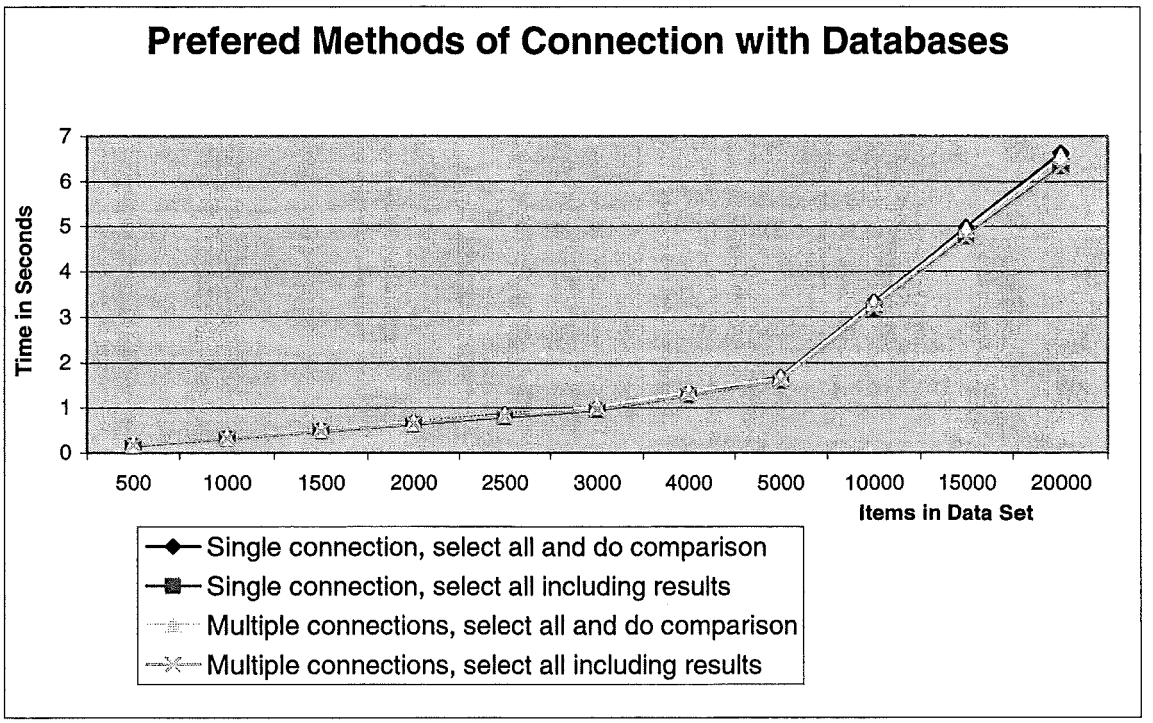


Figure 7: Time Required to Retrieve Datasets Using Four Preferred Methods of Requesting Data from a Database

A series of tests was conducted on the amount of time needed to complete the comparisons if the data is already in memory. Figure 8 shows the results of the analysis. Comparing 20,000 pairs of data takes a little over .12 seconds. This analysis essentially removes the database from the equation. If the information is resident in

memory, it is much more efficient to simply perform the comparisons.

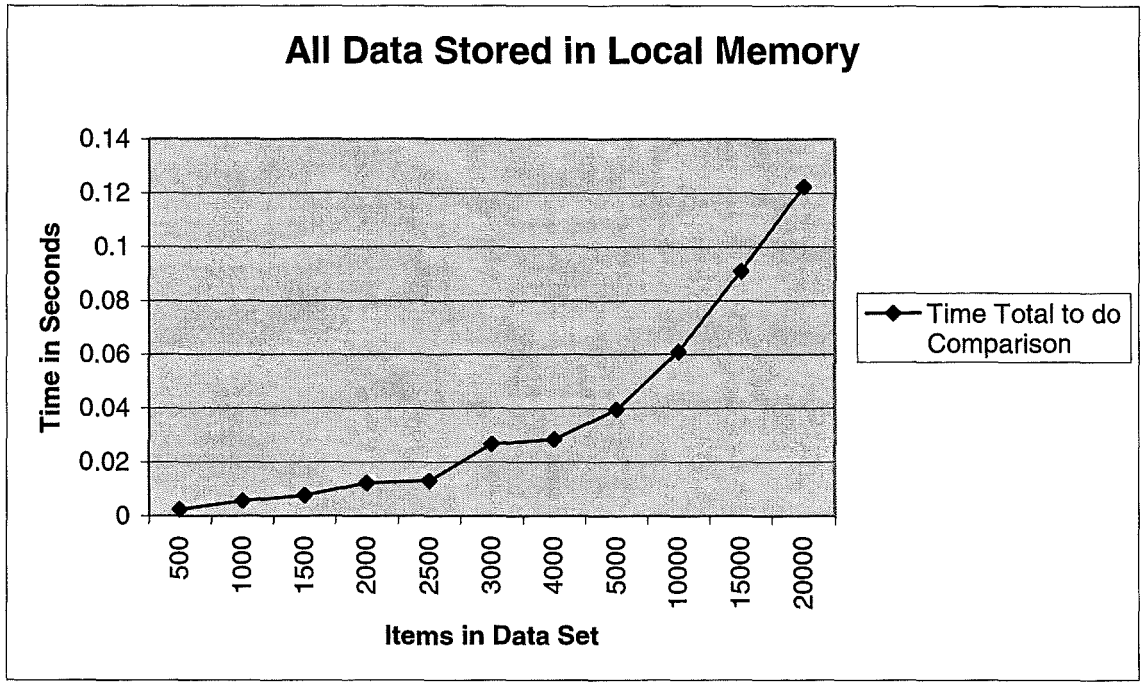


Figure 8: The Amount of Time Needed to Complete a Comparison of Series of Two Data Items when the Data is Resident in Local Memory

Figure 9 shows the times required to perform the four preferred methods with their comparisons and the time to perform the same comparisons when the database is not used. The four methods involving the database take considerably longer to complete the comparisons. It appears that a unification framework that includes a local representation based upon storing the representation in memory will out perform a unification framework based on a database local representation.

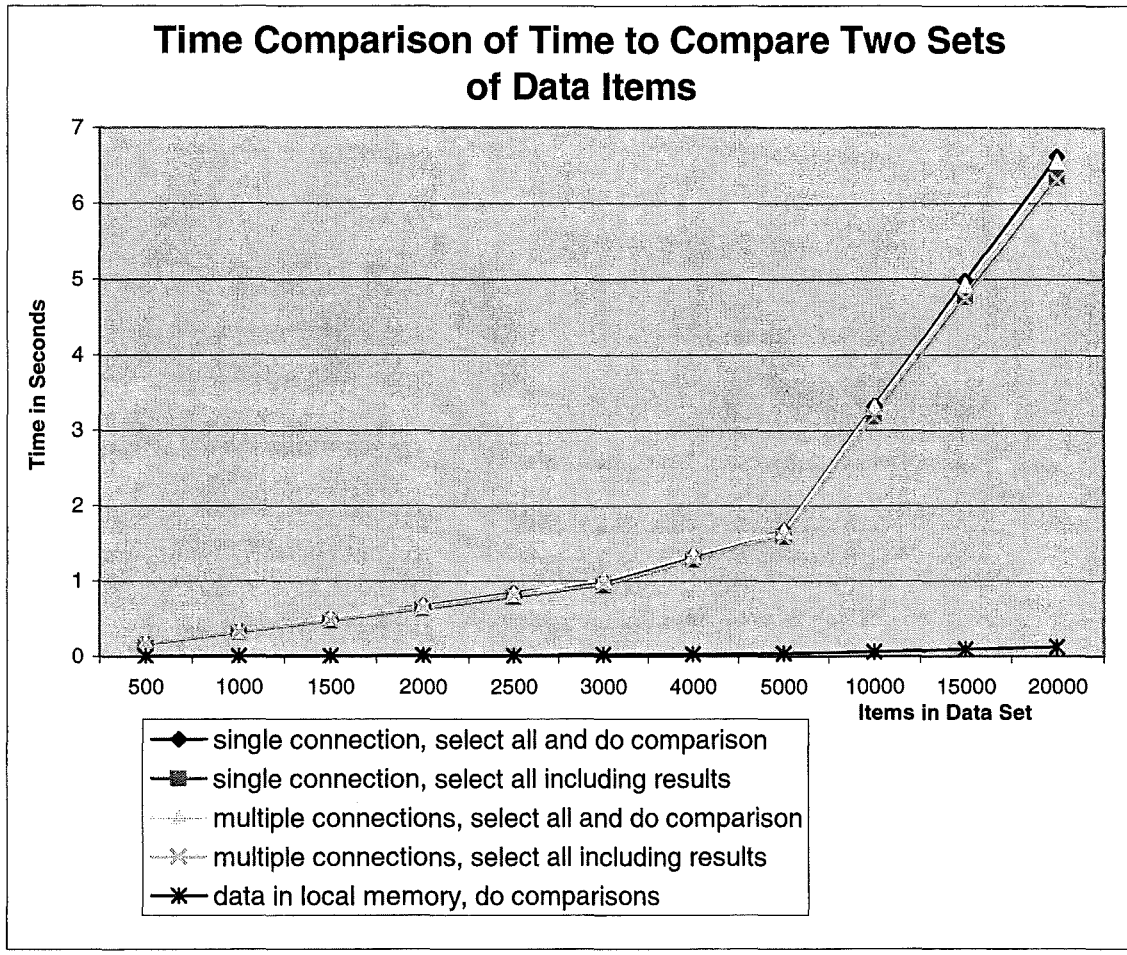


Figure 9: The Four Preferred Methods of Connection and Comparison with the Time Results from the Same Comparisons if the Data is Resident in Local Memory

Additional graphs showing various side by side visualizations of the time required by assorted methods can be found in Appendix H.

Chapter 5

Analysis

The performance of the dbUNiFier was compared to a MOMIS emulation. Both of these pieces connected to the external data sources, retrieved the schema, and simulated the analysis of the schema. Several analysis tests were conducted to evaluate key portions of the dbUNiFier.

5.1 Description

The full analysis component for the dbUNiFier is outlined above in chapter 3. A full analysis piece for the MOMIS emulation was also written. Since the MOMIS framework does not examine instance data, the instance data metric was not included in the MOMIS emulation. Both software used the same wrapper components and retrieved the schema and data in the same manner. The MOMIS documentation does not specify how the data is to be retrieved so for simplicity the same process of collecting the data from the remote data sources was shared by both pieces of software.

To determine the completeness of each unifier, a simple test was devised. Two schema were loaded into the remote data sources. The first had five tables with a total of twenty attributes and the second had ten tables with a total of

forty attributes. In each data source was placed an attribute named 'email'. In the second database was placed an attribute named 'email_address'. In both databases was placed an attribute with the exact same instance data but the attribute names were not related. In the first database, the attribute was named 'fhqwhgads' and in the second, it was named 'strong_bad'. Sample data and schema structure can be found in Appendix I. A second test was devised to see if both or either applications could find matching tables. Tables were replaced in both schema with exact duplicate tables. Both had the same attribute names and instance data. A third test was devised where one attribute's name was changed in one of the duplicate table from the previous test.

The dbUNiFier uses a database for the local representation while the MOMIS emulation uses a memory resident local representation. Since findings outlined in the pre-analysis section of this document show that a memory based application will out perform a database based application, a side by side comparison of the time it takes to perform the analysis of the schema and instance data was not conducted. The MOMIS emulation does not use the exact language and platform defined in MOMIS. The MOMIS unification framework was selected because it was the only true framework in the examined unification methods. It does not examine the contents of the two databases as extensively as the

dbUNiFier can. The MOMIS unifier definition does not examine instance data while dbUNiFier can examine the instance data. Additionally, the metrics modules being used in the analysis are not robust metrics modules that truly represent any specific unification method. The modules examine the data and find matches but are not equipped to find all the matches the various unification methods can find. Since dbUNiFier can include dozens of metrics modules and MOMIS can only use the ones in the MOMIS definition, and since MOMIS does not perform the time intensive examination of the instance date, time comparisons between the two are not very meaningful.

MOMIS uses a memory based local representation of the unified schema and stores attribute information in memory so there is a limit to the size of the data sources that can be unified. Since MOMIS only examines at the attribute level and not the instance level. The amount of data MOMIS needed to process is small compared to an instance evaluating unifier. There is still a limit to the size or combination of schema MOMIS can process. To show that the use of a memory based local representation is inferior to a database in a unification method that is designed to handle the widest variety of data sources, two analysis programs were written to test how the two unification applications would perform as data was added to the local repositories. Both pieces of software loaded a sample set of data into memory.

The sample set was used to simulation data sources of various sizes. The sample set was the same data used for the pre-analysis evaluations. The piece representing the dbUNiFier used a local database. The source code for the dbUNiFier representation can be found in Appendix J. The piece representing the MOMIS application used local memory to store the data. The source code for the MOMIS representation can be found in Appendix K. Approximately 11 megs of data were loaded into each programs data store and 20,000 comparisons were made between random pairs of data. Then, another eleven megabytes of data were loaded into the repository and another 20,000 comparisons were made.

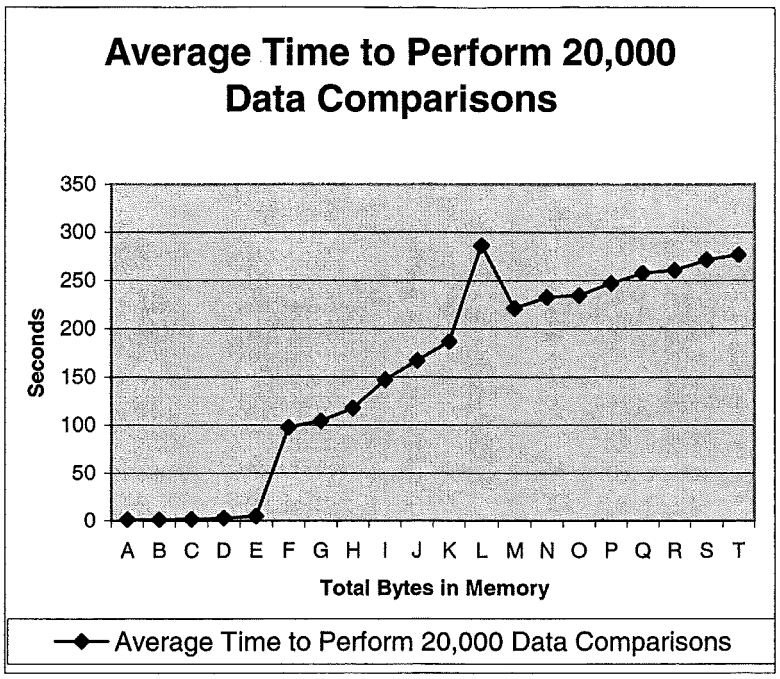
5.2 Results of Analysis

In the first test to see if the two unifiers would identify the three different match types seeded in the schema, the dbUNiFier was able to find all three matches. The application indicated a potential match between the two attributes with the same name. It indicated a potential match between the attributes named 'email' and 'email_address'. It also indicated a potential match between the two attributes with identical instance data. The application based on MOMIS was able to find the attribute matches based on the attribute name but was unable to detect the potential match based on the instance data.

Both applications found the table match in each run of test two. The dbUNiFier found the matching table three times - once for each of the unification metrics included. MOMIS only found the two matches based on the attribute names. The test applications indicated a potential table match by showing all attributes in the table as being potential matches. The logic was not built into either application to truly identify the table as a match but the result data was available if the logic was built into the applications.

On the third test MOMIS, was confused and only found the potential matches based on the attribute names. It did not identify the potential match based on the instance data for the remaining attribute in the table. DbUNiFier, however, was able to identify the two attributes though their names were not identical nor similar.

The last series of tests was to examine the performance of the two unifiers based on the amount of data stored in the local repository. As data was added to the memory resident repository of the MOMIS emulation, the performance of the application degraded. Figure 10 shows the average time needed to evaluate the 20,000 comparisons as data was added to the local memory. This average represents ten runs of the application.



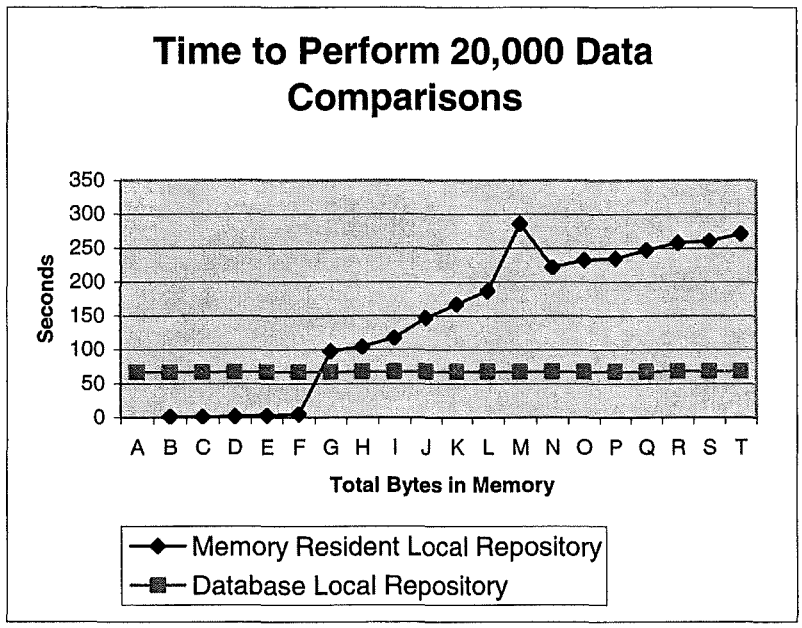
Total Bytes in Memory	
A	11,655,840
B	23,311,680
C	34,967,520
D	46,623,360
E	58,279,200
F	69,935,040
G	81,590,880
H	93,246,720
I	104,902,560
J	116,558,400
K	128,214,240
L	139,870,080
M	151,525,920
N	163,181,760
O	174,837,600
P	186,493,440
Q	198,149,280
R	209,805,120
S	221,460,960
T	233,116,800

Figure 10: Average Comparison Time of MOMIS Emulation as Data was Added to Memory

During the first few iterations of the program, the comparison time was minimal. As more and more data was added to the memory, the performance degraded. A large time spike occurred when 139,870,080 bytes of memory were used to store data. The spike subsided immediately after as more items were added to the memory and degraded continually thereafter. The spike occurred in every test run.

dbUNiFier performance does not degrade as additional data was added to the local representation. The dbUNiFier application was not affected by the amount of items stored. In fact, the time to perform 20,000 comparisons was better

in most instances. This test pulled the data being compared from the database. The dbUNiFier application performed the comparisons on items after they were in memory. Figure 11 shows both the degrading performance of the MOMIS application and the steady performance of the dbUNiFier application.



	Total Bytes in Memory
A	11,655,840
B	23,311,680
C	34,967,520
D	46,623,360
E	58,279,200
F	69,935,040
G	81,590,880
H	93,246,720
I	104,902,560
J	116,558,400
K	128,214,240
L	139,870,080
M	151,525,920
N	163,181,760
O	174,837,600
P	186,493,440
Q	198,149,280
R	209,805,120
S	221,460,960
T	233,116,800

Figure 11: Time to Perform 20,000 Comparisons as Data is Added to both dbUNiFier and MOMIS Local Data Store

The next test was to determine what would happen if the memory was continually loaded with more and more data. The application was modified so that it would not stop after loading memory twenty times with about eleven megabytes of

data between comparisons. After the 45th addition to memory, the application returned a memory error: Out of Memory! The MOMIS emulation was only able to handle 536,168,640 bytes of data before the application could no longer perform its analysis of the data. A second run of the modified application was performed with the same results. Figure 12 shows the performance through both runs of the application to the point that the application prematurely terminated. The time measured to complete the comparisons was considerably different in both runs. The goal of this test was not to examine the time to perform the comparisons but was to determine at what point the application would no longer be able to operate due to memory limitations.

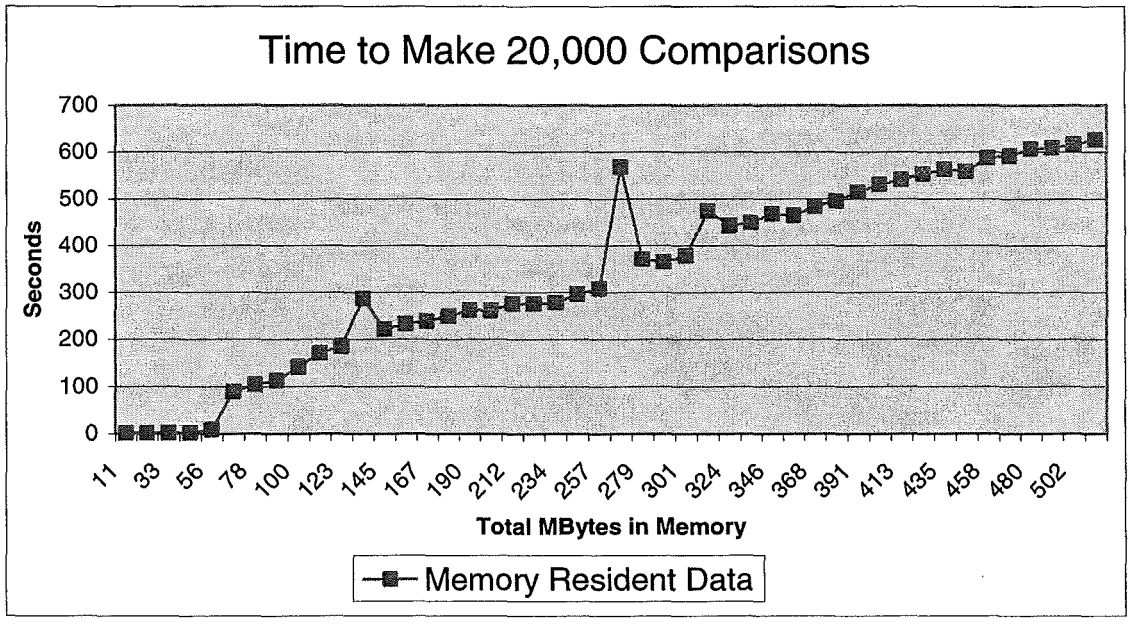


Figure 12: Time Required to Make the Comparisons with Memory Added to the Point the Application Failed

The final test was to determine if dbUNiFier would out perform the MOMIS application when more than 536,168,640 bytes of data were present in the local representation. DbUNiFier continued to perform when more data was present than in MOMIS. 582,792,000 bytes total were loaded into the dbUNiFier. The application continued to be able to perform the analysis and was able to do it in the same amount of time required when less data was in the local database. Figure 13 shows that the dbUNiFier continued to run long after the MOMIS application died due to memory failure.

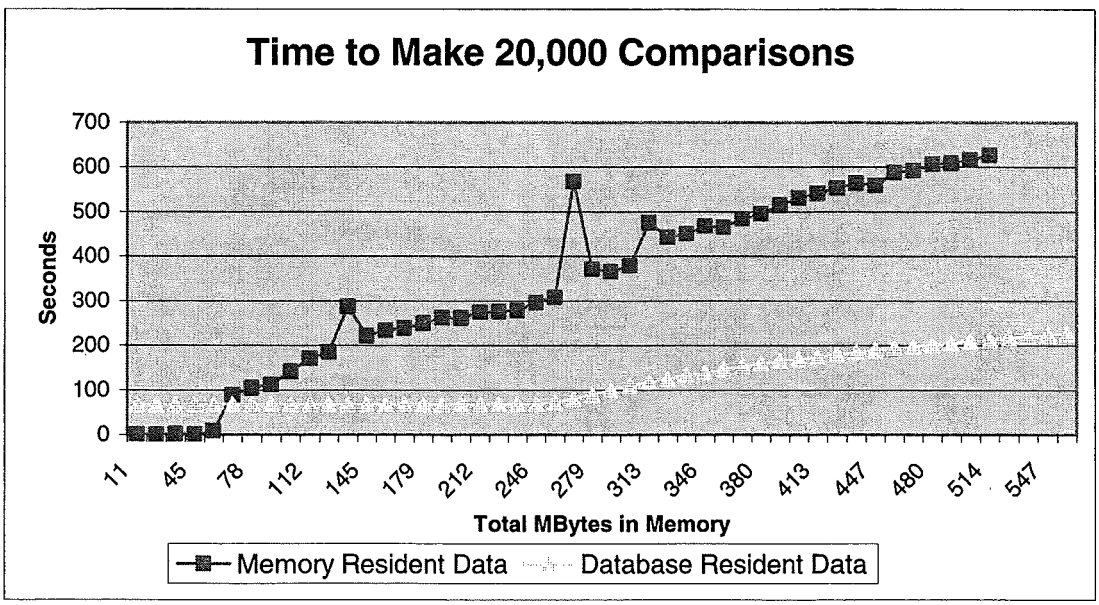


Figure 13: The dbUNiFier Application Continued to Run after the MOMIS Application Halted

Chapter 6

Comparisons

The dbUNiFier framework was designed to be a solution to a wide variety of unification situations. Selection of a previous methodology to compare the dbUNiFer was a difficult task. While the subject of data source unification is rich in research, the area of a structure to tie existing unification methods into a single application is almost nonexistent. There is no research that was found that examines the possibility of tying future unification methods into a framework. Every method has a framework in which it works. Those frameworks are not designed to allow for any expansion. The only method that includes a framework that was designed to encompass an exterior framework is MOMIS. The MOMIS architecture is based on the Artemis unification method.

The MOMIS method was selected after the dbUNiFier framework was envisioned, designed, and implemented. The research into the various unification methods was conducted prior to the design and implementation of the dbUNiFier. The idea of a wrapper to allow the unifier to encapsulate and receive the data source contents came from research previously done

on the MOMIS method. So both unification frameworks are able to interact with the exact same set of data sources. Both can manipulate XML, interact with databases, and receive incoming data from the same sources. Table 5 shows that both methodologies are equivalent in the sources they can unify.

	Relational Databases	Object-Oriented Databases	XML files	Text Documents	Web Structures	Other (future or current)
dbUNiFier	X	X	X	X	X	X
MOMIS	X	X	X	X	X	X

Table 5: Both dbUNiFier and MOMIS can Unify a Variety of Data Sources

After MOMIS was selected to be the subject of the analysis, it was discovered that the basic layout of the structure of both MOMIS and dbUNiFier are similar. The underlying architecture is not. External to the MOMIS wrappers are a data and query manipulation function. These are handled by a layer called ODL, which is only referred to as ODL and is based on an Intelligent Integration of Information (I^3) defined language [Beneventano98]. The MOMIS technical documents specify that the I^3 language is used. dbUNiFier allows for the implementation to be in any language. The ODL acts as a two-way translator between the unification

application and the wrapper. Requests coming from the MOMIS unification application are converted to the proper format and then passed through the wrapper. Data coming into the MOMIS application passes through the wrapper, is scrubbed or converted in the ODL layer, and the data is passed to the application. Figure 14 shows the MOMIS wrapper, the ODL, and the unification program. In the dbUNiFier the data conversion and request formatting is done inside the wrapper. This was designed into the dbUNiFier wrapper to keep any pieces needed to interface with a specific data source within the same object or collection of objects. Figure 15 shows the dbUNiFier wrapper and the unification program.

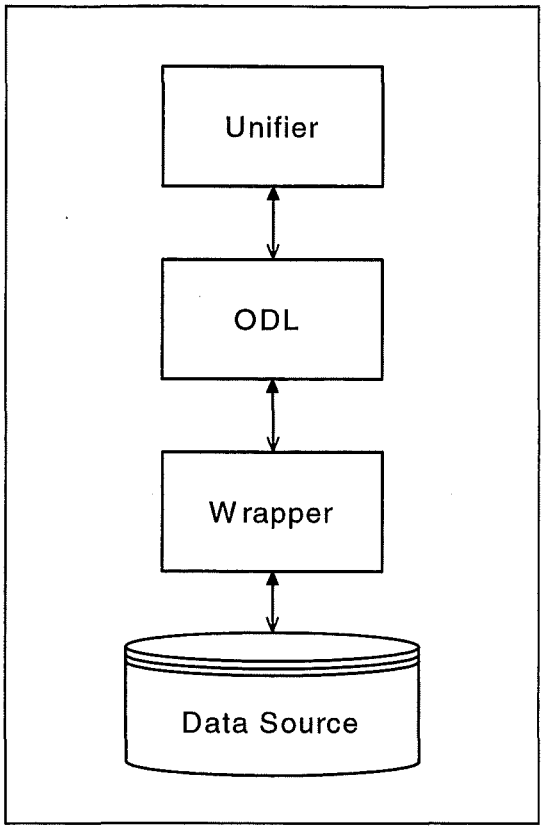


Figure 14: MOMIS Wrapper and Data Translation/Conversion Architecture

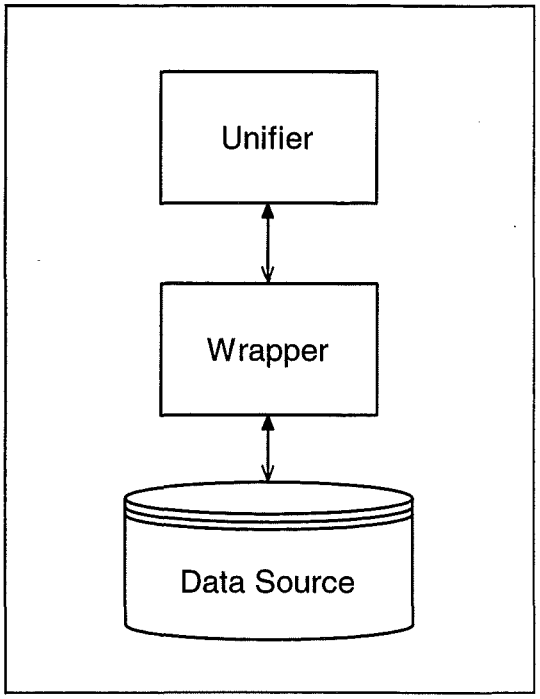


Figure 15: dbUNiFier Wrapper and Data Translation/ Conversion Architecture

MOMIS and dbUNiFier both have internal representations but define the data storage methods differently. MOMIS uses a memory resident graphing scheme to keep track of the data structures being imported. DbUNiFier specifies that a database is to be used for the internal storage. MOMIS required less time over all because of the time overhead to store and retrieve data from the database. Unfortunately tests also showed that the computer used for the analysis was only able to hold under 600 megabytes of data in its

memory structures. The database method performed the actual calculations much more efficiently than the memory overloaded MOMIS. The amount of memory that was used prior to the MOMIS applications inability to perform was based upon the computer used during the tests. All computers have a finite amount of memory and swap memory. Therefore the MOMIS application will always have a performance ceiling. dbUNiFier is not limited by the same ceiling. The overall time needed to perform a unification analysis with dbUNiFier is higher when a small set of data is being analyzed. However, the MOMIS emulation was unable to perform any tasks once the local memory was saturated. DbUNiFier was able to perform the calculations with well over 600 megabytes stored in the database and never experienced the degradation in performance that the MOMIS application experienced.

The MOMIS unifier only determines matches on the attribute level. dbUNiFier allows for the examination of instance data by adding a unification component for that purpose. MOMIS framework interfaces with only one unification subcomponent: the Artemis unification method. While Artemis is a collection of previous unification algorithms, MOMIS does not allow for expansion of new methods of algorithms. dbUNiFier allows for the use of the Artemis set and other unification algorithms as component. Both applications were able to find matching tables but MOMIS was unable to find the matching tables when one attribute was

named differently. Table 6 shows that MOMIS and dbUNiFier are both able to discover matches on the attribute and table level but MOMIS is not equipped to find matches on the instance level.

	attribute level matches	table level matches (based on attribute level matches)	instance level matches	table level matches (based on instance level matches)
dbUNiFier	X	X	X	X
MOMIS	X	X		

Table 6: Both Unification Methods Ability to Find Matches of Different Types

Additional information about both MOMIS and dbUNiFier is shown in table 7. dbUNiFier is language, platform and middle-ware independent while MOMIS is not. Both unifiers examine the data source schema at the attribute level while dbUNiFier also has the ability to examine the data found in the data sources. MOMIS is a memory based application that will terminate if the combined size of the data sources is greater than the memory available to the application. DbUNiFier uses a database to store the data and information from the remote data sources. Both applications can determine logically identical tables based upon the application finding each of the individual attributes that comprise the two tables being examined.

	MOMIS	dbUNiFier
Platform independent		X
Language independent		X
Does not rely on middle-ware		X
Examines attributes (shallow)	X	X
Examines instances (deep)		X
Memory resident repository	X	
Database resident repository		X
Granularity matches	X	X

Table 7: Additional Information about MOMIS and dbUNiFier

Chapter 7

Conclusion

The dbUNiFier framework is a successful generic framework to unify textual data found in remote heterogeneous data sources. The framework takes into account previous findings in the field of schema unification. It allows for connection to a variety of data sources. New data sources can be added easily through the use of wrappers. The dbUNiFier framework allows for schema unification at both the attribute and instance levels. Metrics modules can be added or removed easily. The metrics modules that are part of an implementation of the dbUNiFier can either be selected to be used or not used during a unification attempt.

Previous research has shown that the unification problem can be solved more completely if more than one unification method is used. Previously, these methods could only work together as separate pieces. The data needed for one unification method may be needed for a second or more methods being used. The findings from one method may enhance another method. Previously, these finding would have to be transferred from the originating program to subsequent unification methods. In order for these unification applications to be used together, one application would have to completely finish its evaluation

of the data source schemas before the next application based on a different method would begin to evaluate the data. If the second method finds a single match that the first did not uncover, the first application may need to be run again with the new mapping as additional input. Compound this with a dozen unification methods with half relying on mapping data as input and the result is a very complex process of integrating the applications into a single process.

dbUNiFier fully automates the interactions between multiple unification methods. Inputs for all methods used as unification components are gathered at one time. Each method is run once. The resulting matches are then used as new input for the methods that require more than one pass of their unification algorithm in order to fully evaluate the relationships between the schemas. The methods work together automatically instead of as separate pieces of a multi-step process dependent upon a human to pass new information between the unification applications.

Existing methods working within the dbUNiFier framework can share data easily. The results of one or all metrics modules can be used as input to the rest of the metrics modules being used. The existing and future unification methods work together in an iterative fashion where all methods examine the schemas being integrated. The results

are examined and mappings are accepted or rejected. The findings can be used in the next iteration by all the modules. It would no longer be necessary to complete a run of one unification method and then, begin the run of the next unifier.

The dbUNiFier overcomes the problems with each examined unification methodology as previously outlined. The framework is platform and language independent. It allows for examination of the schema in a variety of methods and also allows for examination of the instance data in a variety of methods. The framework can interface with a variety of sources and the framework is expandable in the unification modules used. The framework removes the rigidity of previous methods and allows the operator to examine the results of the unification analysis on several levels and gives the operator the ability to combine results in a weighted fashion that can be manipulated as needed.

The dbUNiFier framework provides for manageable communication between the application and the external data sources. The idea of a wrapper has been included and has been enhanced with the processes required to convert data from the format of the external source and the format used for the unification applications local repository. This conversion is automated into the wrapper and allows for easy modification of the translations or addition of new data

sources. The design of the wrapper is general enough to allow for addition of future data source types but specific enough to define how the data should be formatted once it leaves the wrapper.

The dbUNiFier can be considered slow when unifying small data sources or being used to only evaluate the attribute level of the schema. On unifications of small sources, the memory-based applications can provide their results faster. When instance level comparison is introduced, or when the data sources have very large schema, the use of a database allows the dbUNiFier application to continue to function when a memory-based unifier runs out of memory.

Additional features have been added that did not exist in previous unification definitions. This does not increase the performance of the framework, but a reports module will help disseminate information found during the unification process. Future uses for the unification applications cannot be predicted so the way the findings will be used cannot be predicted. A reporting module enables the unifier to pass the data to external applications, post new tables to an html document, email a notification that the unification has been completed, or any number of other notifications. No other unification method specifies anything about reporting the findings externally of the application itself.

The dbUNiFier can be used as a client/server, server, or standalone networked application. It can be run as a batched process or as an interactive tool. The resulting mappings table generated by the unification process can be used by outside applications. The results of the unification attempt can be output in a report fashion.

The dbUNiFier is more complete than previous unification methods or frameworks. It can handle extremely large data sources. As a complete unification application with a combination of metrics modules installed, it can provide more matched attributes than the individual modules themselves.

By studying previous unification methodologies and the types of schema they are able to unify, a new unification framework has been successfully devised that is more complete than previous methods and is more expandable than pervious methods. As a combination unifier, the dbUNiFier has the ability to find more matches than single methods.

Chapter 8

Future work

When requesting data from the database, the research presented found the most efficient method was to open a connection, make the request, and close the connection. The connection was established immediately before the request was made. It would be interesting to know if the age of the connection makes a difference. There are several aspects that deserve further analysis in this method of request to find out if this condition holds under other circumstances. Other areas to explore include how performance is affected while handling huge data sets, when very large numbers of requests are needed, with different implementations in various programming languages, with different configurations of the database, or while using other databases or different types of data sources.

Unification of databases is becoming a 'hot topic'. Advances in the field have occurred during the design of the dbUNiFier. The presented framework is designed to work with all conventional data sources and unification methods. The presented framework is also designed to allow interfacing with new sources and include new unification components.

When future expansion was factored in it was based on the current structure of the sources and unification methods. There are characteristics that are common among the sources and these characteristics were factored in a way to allow the expansion. The same is true with the unification methods. The database and database unification fields are very innovative. The characteristics that apply to current technology may not be applicable in future technologies. There is no way to project future changes in the overall database field. Additional research is needed regularly to ensure that dbUNiFier is able to incorporate new unification methods, data sources, structuring, data format conversion, and multiple database querying.

There are a handful of published journal articles that include information about the actual mapping tables used to store the matching attributes. Research into the best possible structure for the mapping table would be beneficial. The mapping table is the result of the unification process. There will be external applications that need the relationships found in this table. Since the nature of these applications will vary, so might the best structure for the mapping table. Applications that perform thousands or even millions of requests in a short period of time are very probable. An efficiency boost in the time needed to access the information in the mapping table will

benefit applications of this nature. Other benefits may be uncovered.

The unification components provide a wide variety of possible projects. Research into specific methods is obvious. The dbUNiFier is designed to use each of the components but research into their workings was to develop an understanding of how they work and how they can work with other unification components. A complete analysis of their inner workings was not conducted. Implementation of each of the unification methods as components to work with the dbUNiFier would be a sizable project.

A cross-platform analysis of the dbUNiFier design would help strengthen the argument that the dbUNiFier framework is the superior method of unifier design. The analysis software used to evaluate the performance of dbUNiFier and MOMIS were written using Perl and a MySQL database. Would the results found using this platform be found if the analysis was performed on another platform? dbUNiFier was designed to be platform independent. The framework design will work with any programming language with the ability to access remote databases. There are many combinations of languages and databases to use for the internal representation.

APPENDIX A

dbUNiFier Test Application

```
#############################################################
# program: dbUNiFier
#
# Description: Implementation of dbUNiFier methodology
#############################################################

# load required classes used by program
use wrappers::marketingbots;
use wrappers::phobos;
use representation;
use methods::attributename;
use methods::attributenamesubstring;
use methods::attributeinstance;

# open a diagnostic file to keep track of what is happening
open(DIAG,">diag.txt");

########################
# create needed objects
########################

$source1 = marketingbots->new();
$source2 = phobos->new();
$localrep= representation->new();

################################################
# load contents and structure of remote databases
################################################

$localrep->create();
&loadremote1;
&loadremote2;
$localrep->cleantables();

################################
# perform unification evaluations
################################

# perform a 1:1 attribute level match
# based on equality of attribute name
$eval1 = attributename->new();
$localrep->{SOURCE}="attributename";
```

```
&evaluate1;

# perform a 1:1 attribute level match
# based on equality of attribute name as a substring
$eval1 = attributenamesubstring->new();
$localrep->{SOURCE}="attributenamesubstring";
&evaluate1;

# perform a 1:1 instance level match
# based on equality of instance level attributes as a set
$eval2 = attributeinstance->new();
$localrep->{SOURCE}="attributeinstance";
&evaluate2;

# close the diagnostic file
close(DIAG);

end;

# get contents of datasource 1
sub loadremote1{

     # pulls attributes from object
     @test = $source1->getattributes();

     # write the attribute info to the diagnostic file
     foreach $line (@test){
          print DIAG "$line\n";
     }

     # move the attributes to the local repository object
     @{$localrep->{ATTRIBS}}=@test;

     # store the attributes in the local repository
     $localrep->addats;

     # get all instances and build tables
     @tables=@{$localrep->{ATTRIBS}};
     foreach $table (@tables){

          # specify which table we're looking inside
          $source1->{TABLE} = "$table";

          # pull instances from source 1 (table)
          @instances = $source1->getinstances();

          # write instances to the diagnostic file
          foreach $line (@instances){
               print DIAG "$line\n";
          }

          # load instances into local repository object
          @{$localrep->{INSTANCES}}=@instances;
```

```
            # store instances in the local repository
            $localrep->addinstances;
        }
}

# get contents of datasource 2
sub loadremote2{
        # pulls attributes from object
        @test = $source2->getattributes();

        # write the attribute info to the diagnostic file
        foreach $line (@test){
            print DIAG "$line\n";
        }

        # move the attributes to the local repository object
        @{$localrep->{ATTRIBS}}=@test;

        # store the attributes in the local repository
        $localrep->addats;

        # get all instances and build tables
        @tables=@{$localrep->{ATTRIBS}};

        # write all instances to the diagnostic file
        foreach $table (@tables){
            print DIAG "$table\n";
        }

        # get all instances and build tables
        foreach $table (@tables){

            # specify which table we're looking inside
            $source2->{TABLE} = "$table";

            # specify which source we are looking in
            $localrep->{SOURCE} = "2"."$table";

            # pull instances from source 2 (table)
            @instance=();
            @instances = $source2->getinstances();

            # write instances to diagnostic file
            foreach $line (@instances){
                print DIAG "$table $line\n";
            }

            # load instances into local repository object
            @{$localrep->{INSTANCES}}=@instances;

            # store instance in local repository object
            $localrep->addinstances;
        }
}
```

```
# perform first type of unification analysis (attribute
level)
sub evaluate1{
     # pull all data needed for the unification analysis
     @allattributes = $localrep->getallattributes;
     @allsources = $localrep->getallsources;
     @alltables = $localrep->getalltables;

     # how many attributes are there total?
     $number=@allsources;

     # identify the first source
     $source1=$allsources[0];

     print "[dbunifier] evaluate1: total attributes to eval
= $number; source1 = $source1\n";

     # loop through once for each attribute in source 1
     for($i=0;$i<$number;$i++){
          # split the attributes into sets by source
          # and load them into the evaluation object
          if($source1==$allsources[$i]){
               push(@{$eval1->{SOURCE1}},$allsources[$i]);
               push(@{$eval1->{TABLE1}},$alltables[$i]);
               push(@{$eval1->{NAME1}},$allattributes[$i]);
          }else{
               push(@{$eval1->{SOURCE2}},$allsources[$i]);
               push(@{$eval1->{TABLE2}},$alltables[$i]);
               push(@{$eval1->{NAME2}},$allattributes[$i]);
          }
     }

     # analyze the attributes previously loaded into object
     $eval1->analyze();

     # load evaluation results into the local repository
     @{$localrep->{FINAL1}}=@{$eval1->{FINAL1}};
     @{$localrep->{FINAL2}}=@{$eval1->{FINAL2}};
     @{$localrep->{RESULT}}=@{$eval1->{RESULT}};

     # store results previous loaded into repository
     $localrep->storeresults();

     # clear memory items in object
     @{$eval1->{SOURCE2}}=();
     @{$eval1->{TABLE2}}=();
     @{$eval1->{NAME2}}=();
     @{$localrep->{FINAL1}}=();
     @{$localrep->{FINAL2}}=();
     @{$localrep->{RESULT}}=();
}
```

```
# perform second type of unification analysis (instance
level)
sub evaluate2{
     # pull all data needed for the unification analysis
     @allattributes = $localrep->getallattributes;
     @allsources = $localrep->getallsources;
     @alltables = $localrep->getalltables;

     # how many attributes are there total?
     $number=@allsources;

     # identify the first source
     $source1=$allsources[0];

     print "[dbunifier] evaluate2: total attributes to eval
= $number; source1 = $source1\n";

     # loop through once for each attribute in source 1
     for($i=0;$i<$number;$i++){
          # split the attributes into sets by source
          # and load them into the evaluation object
          if($source1==$allsources[$i]){
               push(@{$eval2->{SOURCE1}},$allsources[$i]);
               push(@{$eval2->{TABLE1}},$alltables[$i]);
               push(@{$eval2->{NAME1}},$allattributes[$i]);
          }else{
               push(@{$eval2->{SOURCE2}},$allsources[$i]);
               push(@{$eval2->{TABLE2}},$alltables[$i]);
               push(@{$eval2->{NAME2}},$allattributes[$i]);
          }
     }

     print "[dbunifier] evaluate by attribute instances\n";

     # determine size of arrays holding data
     $size1=@{$eval2->{NAME1}};
     $size2=@{$eval2->{NAME2}};

     print "[dbunifier] analyze $size1 x $size2\n";

     # perform analysis on each item
     for($i=0;$i<$size1;$i++){
          # signal repository which item we are evaluating
          $localrep->{ATTRIB}=@{$eval2->{NAME1}}[$i];
          $localrep->{SOURCE}=@{$eval2->{SOURCE1}}[$i];
          $localrep->{TABLE}=@{$eval2->{TABLE1}}[$i];

          # pull the instances on the item we are evaluation
          $localrep->getinstances();

          @{$eval2->{INSTANCES1}}=@{$localrep->{INSTANCES}};
#         @{$eval2->{TABLE1}}=$localrep->{TABLE1}[$i];
#         @{$eval2->{TABLE2}}=$localrep->{TABLE2}[$i];
```

```
          # pull info from evaluation object into local
repository
          for($j=0;$j<$size2;$j++){
                push(@{$localrep->{FINAL1}},@{$eval2-
>{NAME1}}[$i]);
                push(@{$localrep->{FINAL2}},@{$eval2-
>{NAME2}}[$j]);

                $localrep->{ATTRIB2}=@{$eval2->{NAME2}}[$j];
                $localrep->{SOURCE2}=@{$eval2-
>{SOURCE2}}[$j];
                $localrep->{TABLE2}=@{$eval2->{TABLE2}}[$j];

                # get instance data from repository
                $localrep->getinstances2();

                # load evaluation object with the instance
date
                @{$eval2->{INSTANCES2}}=@{$localrep-
>{INSTANCES2}};

                # perform analysis
                $eval2->analyze();
          }
     }
     # move analysis results to the local repository
     @{$localrep->{FINAL1}}=@{$eval2->{FINAL1}};
     @{$localrep->{FINAL2}}=@{$eval2->{FINAL2}};
     @{$localrep->{RESULT}}=@{$eval2->{RESULT}};
     $localrep->{SOURCE} = "attributeinstance";

     # clear unneeded memory items
     $localrep->storeresults();
     @{$eval1->{SOURCE2}}=();
     @{$eval1->{TABLE2}}=();
     @{$eval1->{NAME2}}=();
     @{$localrep->{FINAL1}}=();
     @{$localrep->{FINAL2}}=();
     @{$localrep->{RESULT}}=();
}

end;
```

```
package representation;

#############################################################
# program: dbUNiFier
#
# class: local representation
#
# description: Stores the remote data, structures,
#  and info about the unification process
#############################################################

# load required modules used to access the db
use DBI;
$| = 1;

sub new{
     print "[representation] new representation\n";
     $handle = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker");

     $rep->{BOT}=$handle;
     $rep->{ATTRIBS}=[];
     $rep->{TABLES}=[];
     $rep->{SOURCES}=[];
     $rep->{ATTRIBUTES}=[];
     $rep->{FINAL1}=[];
     $rep->{FINAL2}=[];
     $rep->{RESULT}=[];
     $rep->{ATTRIB}=undef;
     $rep->{ATTRIB2}=undef;
     $rep->{SOURCE}=undef;
     $rep->{SOURCE2}=undef;
     $rep->{TABLE}=undef;
     $rep->{TABLE2}=undef;
     $rep->{INSTANCES}=[];
     $rep->{INSTANCES2}=[];


     bless($rep);

     return($rep);
}

sub create{
     print "\n[representation] create representation\n";
     $sql2="drop table if exists originals";
     $sth = $rep->{BOT}->prepare($sql2);
     $sth->execute or die print "drop table failed
\n($DBI::errstr)<br><br>";

     $sql2="create table originals (id INT NOT NULL
AUTO_INCREMENT,sourcename varchar(50) not null,tablename
varchar(50) not null, attributename varchar(50) not null,
```

```
atype varchar(50), isnull varchar(3) not null,keyholder
varchar(50),defaultval varchar(50),extrastuff
varchar(50),PRIMARY KEY(id))";
     $sth = $rep->{BOT}->prepare($sql2);
     $sth->execute or die print "create originals failed
\n($DBI::errstr)<br><br>";
}

sub addats{
#    print "addats $attribs[0]\n";
#    @tables = "";
#push @attribs, @_;
#    ($attrib) = @_;
     @needed=();
     @list=();
     @attribs=@{$rep->{ATTRIBS}};
     foreach $attrib (@attribs){
          print "[representation] attrib = $attrib\n";
          @indivs = split(/\|/,$attrib);
          $thistable="$indivs[0]"."$indivs[1]";
          print "[representation] thistable = $thistable\n";
          &checkifitexists($thistable);
          &addattribute($thistable,@indivs);
          push(@needed,$indivs[1]);
     }
     $this="";
     foreach $item (@needed){
          if($item ne $this){
               push(@list,$item);
               $this=$item;
          }
     }
     @{$rep->{ATTRIBS}}=@list;
     $rep->{SOURCE}=$thistable;
}

sub cleantables{
  print "\n[representation] clean tables\n";
     $sql="select distinct tablename from originals";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "select distinct tablenames
failed\n($DBI::errstr)<br><br>";
     while (@columns = $sth->fetchrow){
          push(@tables,$columns[1]);
     }

     foreach $table (@tables){
          if($table ne ''){
               $sql="alter table $table drop nonejohnd";
#              print "$sql\n";
               $sth = $rep->{BOT}->prepare($sql);
               $sth->execute or die print "select distinct
tablenames failed\n($DBI::errstr)<br><br>";
          }
```

```
        }
}

sub checkifitexists{
        my ($thistable,$nothing)=@_;
#       print "check if $thistable exists\n";
        $size=@tables;
        $tableexists=0;
        for($i=0;$i<$size;$i++){
                if($tables[$i] eq $thistable){
                        $tableexists=1;
                        $i=$size;
                }
        }
        if(!$tableexists){
                &makenewtable($thistable);
                push(@tables,$thistable);
        }
}

sub makenewtable{
        my ($tablename,$nothing)=@_;
        print "[representation] make new table $tablename\n";
        $sql="drop table if exists $tablename";
        $sth = $rep->{BOT}->prepare($sql);
        $sth->execute or die print "drop table failed
\n($DBI::errstr)<br><br>";

        $sql="create table $tablename (nonejohnd INT)";
        $sth = $rep->{BOT}->prepare($sql);
        $sth->execute or die print "create table failed
\n($DBI::errstr)<br><br>";
}

sub addattribute{
        my ($tablename,@indivs)=@_;
        print "[representation] add attribute $indivs[2]\n";
        $sql="alter table $tablename add $indivs[2] $indivs[3]
";
        if($indivs[5]=~/YES/){
                $sql .= "null ";
        }else{
                $sql .= "not null ";
        }
        $sth = $rep->{BOT}->prepare($sql);
        $sth->execute or die print "drop table failed
\n($DBI::errstr)<br><br>";

        &indexattribute(@indivs);
}

sub indexattribute{
        my (@indivs)=@_;
```

```
     print "[representation] index attribute $indivs[2]
(source $indivs[0], table $indivs[1])\n";

     $sql="insert into originals values (NULL,
'$indivs[0]','$indivs[1]','$indivs[2]','$indivs[3]','$indivs
[4]','$indivs[5]','$indivs[6]','$indivs[7]')";
#    print "$sql\n";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "insert into original failed
\n($DBI::errstr)<br><br>";
}

sub addinstances{
     foreach $instance (@{$rep->{INSTANCES}}){
          print "[representation] add instance $instance\n";
          $instance =~ s/\'/\\\'/;
          @items = split(/\|/,$instance);
          $sql="insert into $rep->{SOURCE} values (null,";
          $size=@items;
          print "[representation] size= $size\n";
          for($i=0;$i<$size;$i++){
               $sql.="'$items[$i]'";
               if($i<$size-1){
                    $sql.=",";
               }else{
                    $sql.=")";
               }
          }
          print "$sql\n";
          $sth = $rep->{BOT}->prepare($sql);
          $sth->execute or die print "add instances
failed\n($DBI::errstr)<br><br>";

     }
     $rep->{INSTANCES}=();
}

sub getallattributes{
     @{$rep->{SOURCES}}=();
     @{$rep->{TABLES}}=();
     @{$rep->{ATTRIBUTES}}=();
     $sql = "select * from originals";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "get all attributes
failed\n($DBI::errstr)<br><br>";

     while ( @columns = $sth->fetchrow) {
          push(@{$rep->{SOURCES}},$columns[1]);
          push(@{$rep->{TABLES}},$columns[2]);
          push(@{$rep->{ATTRIBUTES}},$columns[3]);
     }
     return(@{$rep->{ATTRIBUTES}});
}
```

```
sub getallsources{
     return(@{$rep->{SOURCES}});
}

sub getalltables{
     return(@{$rep->{TABLES}});
}

sub storeresults{
  print "[representation] store results $rep->{SOURCE}\n";
     $sql="drop table if exists $rep->{SOURCE}";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "drop table failed
\n($DBI::errstr)<br><br>";

     $sql = "create table $rep->{SOURCE} (item1 varchar(50),
item2 varchar(50), result int)";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "create table $rep->{SOURCE}
failed\n($DBI::errstr)<br><br>";

     $size = @{$rep->{RESULT}};

     for($i=0;$i<$size;$i++){
          print "[representation] inserting $i\n";
          $sql = "insert into $rep->{SOURCE} values ('$rep-
>{FINAL1}[$i]','$rep->{FINAL2}[$i]','$rep->{RESULTS}')";
          $sth = $rep->{BOT}->prepare($sql);
          $sth->execute or die print "create table $rep-
>{SOURCE} failed\n($DBI::errstr)<br><br>";
     }
     $rep->{FINAL1}=();
     $rep->{FINAL2}=();
     $rep->{RESULT}=();
}

sub getinstances{
     $localtable = "$rep->{SOURCE}"."$rep->{TABLE}";
     $sql = "select $rep->{ATTRIB} from $localtable order by
$rep->{ATTRIB} ASC";
     $sth = $rep->{BOT}->prepare($sql);
     $sth->execute or die print "get instances $rep-
>{ATTRIB} from $localtable failed\n($DBI::errstr)<br><br>";
     while ( @columns = $sth->fetchrow) {
          push(@{$rep->{INSTANCES1}},$columns[0]);
    print "[representation] get instance 1 $columns[0]\n";
     }
}

sub getinstances2{
     $localtable = "$rep->{SOURCE2}"."$rep->{TABLE2}";
     $sql = "select $rep->{ATTRIB2} from $localtable order
by $rep->{ATTRIB2} ASC";
     $sth = $rep->{BOT}->prepare($sql);
```

```
     $sth->execute or die print "get instances $rep-
>{ATTRIB2} from $localtable failed\n($DBI::errstr)<br><br>";
     while ( @columns = $sth->fetchrow) {
          push(@{$rep->{INSTANCES2}},$columns[0]);
    print "[representation] get instance 2 $columns[0]\n";
     }
}
```

```
package attributeinstance;

############################################################
# program: dbUNiFier
#
# class: attribute instances
#
# description: Performs the analysis of the databases
#  at the instance level
############################################################

$| = 1;

sub new{
     print "[attributeinstance] new attribute instance
object\n";

     $attributes->{SOURCE1}=[];
     $attributes->{TABLE1}=[];
     $attributes->{NAME1}=[];
     $attributes->{INSTANCES1}=[];

     $attributes->{SOURCE2}=[];
     $attributes->{TABLE2}=[];
     $attributes->{NAME2}=[];
     $attributes->{INSTANCES2}=[];

  $attributes->{SCORE}=0;
     $attributes->{FINAL1}=[];
     $attributes->{FINAL2}=[];
     $attributes->{RESULT}=[];

     bless($attributes);

     return($attributes);
}

sub analyze{
  $score=0;
  $size1=@{$attributes->{INSTANCES2}};
  $size2=@{$attributes->{INSTANCES1}};
  if(($size1==$size2)&&($size1>0)){
    for($i=0;$i<$size;$i++){
      if(@{$attributes->{INSTANCES2}}[$i] == @{$attributes-
>{INSTANCES1}}[$i]){
        $score++;
      }
      $temp = @{$attributes->{INSTANCES2}}[$i];
      $temp2 = @{$attributes->{INSTANCES1}}[$i];
      print "[attributeinstance] 2 = $temp; 3 = $temp2; size
= $size; score = $score\n";
    }
    $score=$score/$size1;
  }else{
```

```
    $score=0;
  }
  print "[attributeinstance] final score = $score\n\n";
     push(@{$attributes->{RESULT}},$score);
}
```

```
package attributename;

#############################################################
# program: dbUNiFier
#
# class: attribute name
#
# description: Performs the evaluation at an attribute
# level by comparing the attribute names for equality
#############################################################

$| = 1;

sub new{
     print "[attributename] new attribute name\n";

     $attributes->{SOURCE1}=[];
     $attributes->{TABLE1}=[];
     $attributes->{NAME1}=[];
     $attributes->{SOURCE2}=[];
     $attributes->{TABLE2}=[];
     $attributes->{NAME2}=[];
     $attributes->{FINAL1}=[];
     $attributes->{FINAL2}=[];
     $attributes->{RESULT}=[];

     bless($attributes);

     return($attributes);
}

sub analyze{
     print "\n[attributename] evaluate by attribute name\n";

     @one=@{$attributes->{NAME1}};
     @two=@{$attributes->{NAME2}};

     $size1=@one;
     $size2=@two;

     print "[attributename] analyze $size1 x $size2\n";

     for($i=0;$i<$size1;$i++){
          for($j=0;$j<$size2;$j++){
               push(@{$attributes->{FINAL1}},$one[$i]);
               push(@{$attributes->{FINAL2}},$two[$j]);
               if($one[$i] eq $two[$j]){
                    push(@{$attributes->{RESULT}},1);
               }else{
                    push(@{$attributes->{RESULT}},0);
               }
          }
     }
     $size=@{$attributes->{RESULT}};
```

```
        print "[attributename] size of results array $size\n";
}
```

```
package attributenamesubstring;

############################################################
# program: dbUNiFier
#
# class: attribute name substring
#
# description: Performs the evaluation at the attribute
# level by checking to see if either attribute is
# a substring of the second attribute in a comparison
# pair.
############################################################

$| = 1;

sub new{
     print "[attributenamesubstring] new attribute name
substring object\n";

     $attributes->{SOURCE1}=[];
     $attributes->{TABLE1}=[];
     $attributes->{NAME1}=[];
     $attributes->{SOURCE2}=[];
     $attributes->{TABLE2}=[];
     $attributes->{NAME2}=[];
     $attributes->{FINAL1}=[];
     $attributes->{FINAL2}=[];
     $attributes->{RESULT}=[];

     bless($attributes);

     return($attributes);
}

sub analyze{
     print "\n[attributenamesubstring] evaluate by attribute
name substrings\n";

     @one=@{$attributes->{NAME1}};
     @two=@{$attributes->{NAME2}};

     $size1=@one;
     $size2=@two;

     print "[attributenamesubstring] analyze $size1 x
$size2\n";

     for($i=0;$i<$size1;$i++){
          for($j=0;$j<$size2;$j++){
               push(@{$attributes->{FINAL1}},$one[$i]);
               push(@{$attributes->{FINAL2}},$two[$j]);
               if(($one[$i] =~ /$two[$j]/)||($two[$j] =~
/$one[$i]/)){
                    push(@{$attributes->{RESULT}},1);
```

```
                }else{
                    push(@{$attributes->{RESULT}},0);
                }
            }
        }
        $size=@{$attributes->{RESULT}};
        print "[attributenamesubstring] size of results array
is $size\n";
}
```

```
package marketingbots;

#############################################################
# program: dbUNiFier
#
# class: marketingbots wrapper
#
# description: Allows communication with the remote
#  data source.
#############################################################

# load required modules used to send requests across HTTP
use LWP::UserAgent;
use HTTP::Request::Common;
$| = 1;

sub new{
     ## create bot
     $bot = new LWP::UserAgent;
     ## define timeout
     $bot->timeout(30);

     $wrapper->{BOT}=$bot;
     $wrapper->{TABLE}=undef;

     bless($wrapper);

     return($wrapper);
}

sub getattributes{
     $page="http://www.marketingbots.com/thesis/mysql_wrappe
r_server-side.cgi?command=1";
     @final = &bot;

     return(@final);
}

sub getinstances{
     $page="http://www.marketingbots.com/thesis/mysql_wrappe
r_server-side.cgi?command=2&table=$wrapper->{TABLE}";
print "$page";
     @final = &bot;

     return(@final);
}

### bot used to pull remote document via HTTP
sub bot($wrapper){
     ## make network request
     $response = $wrapper->{BOT}->request( GET $page );
     ## remove $content from bot structure
     $content = $response->content;
```

```
        ## break response by line into array
        @one = split(/\n/,$content);

        return(@one);
}
```

```
package phobos;

#############################################################
# program: dbUNiFier
#
# class: Wrapper component to Phobos (Oracle) database
#
# description: Allows dbUNiFier to communicate with Phobos.
#############################################################

# load required modules used to access the db
use DBI;
use DBD::Oracle;
$| = 1;

# create a new wrapper object
sub new{
     # login information
     $host="phobos.cocse.unf.edu";
     $user="johnd";
     $passwd="abbassi";

     # check tnsname.ora to find info
     # establish connection with phobos
     $dbh = DBI->connect("dbi:Oracle:host=$host;sid=ORCL",
$user, $passwd);

     # store pointers in memory of object
     $wrapper->{BOT}=$dbh;
     $wrapper->{TABLE}=undef;

     # allow contents of this object to be passed
     bless($wrapper);

     return($wrapper);
}

# returns a list of user attributes for all tables
sub getattributes{

     # select database information from database
     $sql = "select * from COLS";
     print "$sql\n\n";
     $sth = $wrapper->{BOT}->prepare($sql);
     $sth->execute or die print "No info in database
\n($DBI::errstr)<br><br>";

     # pull the response
     while (@columns = $sth->fetchrow) {
          $type=$columns[2];

          # convert anything that needs to be represented
          # differently in the local representation
          if($type =~ /NUMBER/){
```

```
                $type = "int";
            }elsif($type =~ /VARCHAR2/){
                $type = "varchar";
            }

            if($columns[8]=~/Y/){
                $nullvalue="YES";
            }else{
                $nullvalue="NO";
            }

            # format the information to be stored in object

            $line="2|$columns[0]|$columns[1]|$type($columns[5)
            |$nullvalue||$columns[11]|";

            # keep all lines in an array for easy management
            push(@attributes,$line);
        }

        # send that array back to the main program
        return(@attributes);
}

# pulls instances of a specific attribute
sub getinstances{
        @instances=();
        print "get instances from $wrapper->{TABLE}\n";
        $sql = "select * from $wrapper->{TABLE}";
        print "$sql\n\n";
        $sth = $wrapper->{BOT}->prepare($sql);
        $sth->execute or die print "No info in database
\n($DBI::errstr)<br><br>";
        @instance=();

        # pull responses from database
        while ( @columns = $sth->fetchrow) {
            $size=@columns;
            $line="";
            # format the response to the local representation
            for($i=0;$i<$size;$i++){
                $line .= "$columns[$i]";
                if($i<$size-1){
                    $line .= "|";
                }
            }
            # keep track of all instances in an array
            push(@instances,"$line");
        }

        # send that array to the main application
        return(@instances);
}
```

```
#!/usr/bin/perl

############################################################
# program: Server side portion of distributed mySQL wrapper
#
# description: Wrapper component of dbUNiFier calles this
# free standing application to acquire contents of mySQL
# database behind security.  This wrapper portion pulls
# the data on the server and formats it for transport to
# dbUNiFier.
############################################################

# load required modules used to access the db
use DBI;
use CGI::Carp qw(fatalsToBrowser);
use CGI qw (:standard);

# this program acts as a document so we have to send a HTML
header
print "Content-type: text/html\n\n";

# pull parameters from call to this application
$q = new CGI;
@stuff = $q->param;
foreach $stuff (@stuff) {${$stuff} = $q->param($stuff);}

# Get the input
$in = $ENV{'QUERY_STRING'};
@sin = split(/&/,$in);

# sort the input
@commandtemp=split(/=/,$sin[0]);

$command=$commandtemp[1];

# determine what the program should do based on call to
application
if($command eq "1"){
     &connect;
     &query;
     &disconnect;
}
elsif($command eq "2"){
     &connect;
     &query2;
     &disconnect;
}
elsif($command eq "3"){
     &connect;
     &query3;
     &disconnect;
}
else
{
```

```
        &invalid;
        print "command = '$command'\n";
}

end;

### simple connection string
sub connect{
        $dbh = DBI-
>connect("DBI:mysql:marketin_links","marketin_links","slacke
r")or die print "$Mysql::db_errstr<br><br>";
}

### sample query
sub query{
        $sql = "show tables";
        $sth = $dbh->prepare($sql);
        $sth->execute or die print "Query failed
\n($DBI::errstr)<br><br>";

        print "<-- begin show tables -->\n";

        while ( @columns = $sth->fetchrow) {
                print "$columns[0]\n";
        }

        print "<-- end show tables -->\n";

}

### sample query
sub query2{
        $sql = "show columns from EMAILS";
        $sth = $dbh->prepare($sql);
        $sth->execute or die print "Query failed
\n($DBI::errstr)<br><br>";

        print "<-- begin show columns -->\n";

        while ( @columns = $sth->fetchrow) {
                print
"$columns[0]|$columns[1]|$columns[2]|$columns[3]|$columns[4]
\n";
        }

        print "<-- end show columns -->\n";

}

### sample query
sub query3{
        $sql = "select * from EMAILS limit 100";
        $sth = $dbh->prepare($sql);
```

```
     $sth->execute or die print "Query failed
\n($DBI::errstr)<br><br>";

     print "<-- begin show data -->\n";

     while ( @columns = $sth->fetchrow) {
          print
"$columns[0]|$columns[1]|$columns[2]|$columns[3]|$columns[4]
\n";
     }

     print "<-- end show columns -->\n";

}

### disconnection string
sub disconnect{
     $dbh->disconnect;
}

### invalid command
sub invalid{
     print "INVALID COMMAND";
}
```

APPENDIX B

Sample Run of dbUNiFier

```
[representation] new representation

[representation] create representation
[representation] attrib = 1|EMAILS|email|varchar(70)||||
[representation] thistable = 1EMAILS
[representation] make new table 1EMAILS
[representation] add attribute email
[representation] index attribute email (source 1, table
EMAILS)
[representation] attrib = 1|EMAILS|url|varchar(70)||||
[representation] thistable = 1EMAILS
[representation] add attribute url
[representation] index attribute url (source 1, table
EMAILS)
[representation] attrib = 1|EMAILS|title|varchar(70)||||
[representation] thistable = 1EMAILS
[representation] add attribute title
[representation] index attribute title (source 1, table
EMAILS)
[representation] attrib = 1|EMAILS|section|int(1)|||0|
[representation] thistable = 1EMAILS
[representation] add attribute section
[representation] index attribute section (source 1, table
EMAILS)
[representation] attrib =
1|EMAILS|timestamp|timestamp(14)|YES|||
[representation] thistable = 1EMAILS
[representation] add attribute timestamp
[representation] index attribute timestamp (source 1, table
EMAILS)
http://www.marketingbots.com/thesis/mysql_wrapper_server-
side.cgi?command=2&table=EMAILS[representation] add instance
johan_casper@yahoo.co.uk|http://www.artzweb.net/webhosting/c
ompare_plans.php|Compare ARTzWeb.net web hosting
services|2|20030218011051
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.artzweb.net/web
hosting/compare_plans.php','Compare ARTzWeb.net web hosting
services','2','20030218011051')
```

```
[representation] add instance
johan_casper@yahoo.co.uk|http://www.distractacom.com|Distrac
tacom network of Internet sites.|1|20030218011354
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.distractacom.co
m','Distractacom network of Internet
sites.','1','20030218011354')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.artzweb.net/webhosting/f
ree_uk_dialup.php|Free ISP services to all UK based
customers|2|20030218011650
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.artzweb.net/web
hosting/free_uk_dialup.php','Free ISP services to all UK
based customers','2','20030218011650')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.westhamfans.org/articles
/transfer_talk.htm|West-ham transfer talk|6|20030218011951
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.westhamfans.org
/articles/transfer_talk.htm','West-ham transfer
talk','6','20030218011951')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.artzweb.net/webhosting/s
mall_business_hosting_service.php|Small business web hosting
service.|1|20030218012253
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.artzweb.net/web
hosting/small_business_hosting_service.php','Small business
web hosting service.','1','20030218012253')
[representation] add instance
abstract35@hotmail.com|http://www.adminder.com/c.cgi?abstrac
t&worldsub|How to Generate Guaranteed Traffic to Your
Opportunity|1|20030218012331
[representation] size= 5
insert into 1EMAILS values
(null,'abstract35@hotmail.com','http://www.adminder.com/c.cg
i?abstract&worldsub','How to Generate Guaranteed Traffic to
Your Opportunity','1','20030218012331')
[representation] add instance
johan_casper@yahoo.co.uk|http://adsaturation.com/cgi-
bin/sa/d.cgi/FL9450/links.html|Submit your site to 250 00
ffa sites free|2|20030218012556
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://adsaturation.com/cg
i-bin/sa/d.cgi/FL9450/links.html','Submit your site to 250
00 ffa sites free','2','20030218012556')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.artzweb.net/webhosting/s
```

```
tarter_hosting_service.php|Starter web hosting service pack!
Only 9.99 euros per month|2|20030218012903
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.artzweb.net/web
hosting/starter_hosting_service.php','Starter web hosting
service pack! Only 9.99 euros per
month','2','20030218012903')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.artzweb.net/domainname/d
omain_name_registration.php|Domain name registration
services. Register your domain name|2|20030218013201
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.artzweb.net/dom
ainname/domain_name_registration.php','Domain name
registration services. Register your domain
name','2','20030218013201')
[representation] add instance
johan_casper@yahoo.co.uk|http://www.namesatnames.net|namesat
names - Domain names and web hosting
services.|2|20030218013454
[representation] size= 5
insert into 1EMAILS values
(null,'johan_casper@yahoo.co.uk','http://www.namesatnames.ne
t','namesatnames - Domain names and web hosting
services.','2','20030218013454')
select * from COLS

[representation] attrib = 2|EMPLOYEES|ID|int(22)|NO|||
[representation] thistable = 2EMPLOYEES
[representation] make new table 2EMPLOYEES
[representation] add attribute ID
[representation] index attribute ID (source 2, table
EMPLOYEES)
[representation] attrib =
2|EMPLOYEES|NAME|varchar(128)|YES|||
[representation] thistable = 2EMPLOYEES
[representation] add attribute NAME
[representation] index attribute NAME (source 2, table
EMPLOYEES)
[representation] attrib =
2|EMPLOYEES|TITLE|varchar(128)|YES|||
[representation] thistable = 2EMPLOYEES
[representation] add attribute TITLE
[representation] index attribute TITLE (source 2, table
EMPLOYEES)
[representation] attrib = 2|EMPLOYEES|PHONE|CHAR(8)|YES|||
[representation] thistable = 2EMPLOYEES
[representation] add attribute PHONE
[representation] index attribute PHONE (source 2, table
EMPLOYEES)
[representation] attrib =
2|PROFILE|USERID|varchar(20)|YES|||
```

```
[representation] thistable = 2PROFILE
[representation] make new table 2PROFILE
[representation] add attribute USERID
[representation] index attribute USERID (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|REALNAME|varchar(60)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute REALNAME
[representation] index attribute REALNAME (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|ADDRESS|varchar(60)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute ADDRESS
[representation] index attribute ADDRESS (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|CITY|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute CITY
[representation] index attribute CITY (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|STATE|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute STATE
[representation] index attribute STATE (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|ZIP|varchar(20)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute ZIP
[representation] index attribute ZIP (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|COUNTRY|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute COUNTRY
[representation] index attribute COUNTRY (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|PHONE|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute PHONE
[representation] index attribute PHONE (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|FAX|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute FAX
[representation] index attribute FAX (source 2, table
PROFILE)
```

```
[representation] attrib =
2|PROFILE|EMAIL|varchar(100)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute EMAIL
[representation] index attribute EMAIL (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|AOLIM|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute AOLIM
[representation] index attribute AOLIM (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|ICQ|varchar(30)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute ICQ
[representation] index attribute ICQ (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|YAHOOIM|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute YAHOOIM
[representation] index attribute YAHOOIM (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|MSN|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute MSN
[representation] index attribute MSN (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|EXTRA1|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute EXTRA1
[representation] index attribute EXTRA1 (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|EXTRA2|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute EXTRA2
[representation] index attribute EXTRA2 (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|EXTRA3|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute EXTRA3
[representation] index attribute EXTRA3 (source 2, table
PROFILE)
[representation] attrib =
2|PROFILE|EXTRA4|varchar(40)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute EXTRA4
[representation] index attribute EXTRA4 (source 2, table
PROFILE)
```

```
[representation] attrib =
2|PROFILE|STATUS|varchar(20)|YES||NULL|
[representation] thistable = 2PROFILE
[representation] add attribute STATUS
[representation] index attribute STATUS (source 2, table
PROFILE)
[representation] attrib = 2|T3|A|int(22)|YES|||
[representation] thistable = 2T3
[representation] make new table 2T3
[representation] add attribute A
[representation] index attribute A (source 2, table T3)
[representation] attrib = 2|T3|B|int(22)|YES|||
[representation] thistable = 2T3
[representation] add attribute B
[representation] index attribute B (source 2, table T3)
get instances from EMPLOYEES
select * from EMPLOYEES

get instances from PROFILE
select * from PROFILE

[representation] add instance 1|old username and
password|||||||oldaccount@9000freeleads.net||||||||APPROVE
D
[representation] size= 19
insert into 2PROFILE values (null,'1','old username and
password','','','','','','','','oldaccount@9000freeleads.net
','','','','','','','','','APPROVED')
[representation] add instance
55|admin||||||||||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'55','admin','','','','','','','','','','','','','','
','','','APPROVED')
[representation] add instance
56|admin||||||||||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'56','admin','','','','','','','','','','','','','','
','','','APPROVED')
[representation] add instance
57|admin||||||||||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'57','admin','','','','','','','','','','','','','','
','','','APPROVED')
[representation] add instance
58|admin||||||||||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'58','admin','','','','','','','','','','','','','','
','','','APPROVED')
[representation] add instance
60|none|||||||none@9000freeleads.net||||||||APPROVED
```

```
[representation] size= 19
insert into 2PROFILE values
(null,'60','none','','','','','','','none@9000freeleads.n
et','','','','','','','','','APPROVED')
[representation] add instance
59|admin||||||||||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'59','admin','','','','','','','','','','','','','','
','','','APPROVED')
[representation] add instance 61|Mike|PO Box 1265|Kill Devil
Hills|NC|27948|USA|1-877-843-4609|1-877-843-
4609|eaglenet25@ureach.com||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values (null,'61','Mike','PO Box
1265','Kill Devil Hills','NC','27948','USA','1-877-843-
4609','1-877-843-
4609','eaglenet25@ureach.com','','','','','','','','','APPRO
VED')
[representation] add instance
62|Nathaniel|Walters|Miami|FL|33162|U.S.A.|||nwalters1978@ho
tmail.com||||nwalters589|||||APPROVED
[representation] size= 19
insert into 2PROFILE values
(null,'62','Nathaniel','Walters','Miami','FL','33162','U.S.A
.','','','nwalters1978@hotmail.com','','','','nwalters589','
','','','','APPROVED')
[representation] add instance 63|Heather Proud|16 Three Hill
View|Glastonbury|Somerset|BA6
8AU|UK|||ozzyheather@yahoo.com||||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'63','Heather Proud','16
Three Hill View','Glastonbury','Somerset','BA6
8AU','UK','','','ozzyheather@yahoo.com','','','','','','','','
','','CANCELLED')
[representation] add instance 64|Ged Matthews|Glendale,
Satley|Bishop Auckland|County durham|DL13 4HT|United
kingdom|+440191
3504979||ged@choices4u.co.uk||||GednKev@choices4U.co.uk|||||
CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'64','Ged
Matthews','Glendale, Satley','Bishop Auckland','County
durham','DL13 4HT','United kingdom','+440191
3504979','','ged@choices4u.co.uk','','','','GednKev@choices4
U.co.uk','','','','','CANCELLED')
[representation] add instance 65|david cryer|6315 solandra
dr|jacksonville|fla|32210|usa|904-
7774981||crye1217@msn.com|||dcc272003||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'65','david cryer','6315
solandra dr','jacksonville','fla','32210','usa','904-
7774981','','crye1217@msn.com','','','dcc272003','','','','','
','','CANCELLED')
```

```
[representation] add instance 69|Dietmar Heinrich|Borova
536|Pardubice|n/a|53351|Czech
Republic|00420466531337||dietmar215@yahoo.de|||||||||APPROVE
D
[representation] size= 19
insert into 2PROFILE values (null,'69','Dietmar
Heinrich','Borova 536','Pardubice','n/a','53351','Czech
Republic','00420466531337','','dietmar215@yahoo.de','','','',
'','','','','','','APPROVED')
[representation] add instance 70|Don|Box
14335|Spokane|WA|99214|USA|||bidderssweet@yahoo.com|||||||||
CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'70','Don','Box
14335','Spokane','WA','99214','USA','','','bidderssweet@yaho
o.com','','','','','','','','','','CANCELLED')
[representation] add instance 71|Connie Blango|312 Missouri
Avenue, NW|Washington|DC|20011|USA|(202)882-
1578||cblango@aol.com|||||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'71','Connie Blango','312
Missouri Avenue,
NW','Washington','DC','20011','USA','(202)882-
1578','','cblango@aol.com','','','','','','','','','CANCELLE
D')
[representation] add instance 72|bill demarzo|19 orangeburg
rd|old tappan|new
jersey|07675|usa|2017675431||demarzob@bellatlantic.net||||||
|||EXPIRED
[representation] size= 19
insert into 2PROFILE values (null,'72','bill demarzo','19
orangeburg rd','old tappan','new
jersey','07675','usa','2017675431','','demarzob@bellatlantic
.net','','','','','','','','','','EXPIRED')
[representation] add instance 73|John Tanner|3303
LaSalle|Alton|Illinois|62002|USA|||tanman@ezl.com|||||||||CA
NCELLED
[representation] size= 19
insert into 2PROFILE values (null,'73','John Tanner','3303
LaSalle','Alton','Illinois','62002','USA','','','tanman@ezl.
com','','','','','','','','','','CANCELLED')
[representation] add instance 74|Art Hogenbirk|105 Nerepis
Road|Grandbay/Westfield|NB|E5K 2Z1|Canada|506 757
2246||arthogenbirk@sprint.ca|||||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'74','Art Hogenbirk','105
Nerepis Road','Grandbay/Westfield','NB','E5K
2Z1','Canada','506 757
2246','','arthogenbirk@sprint.ca','','','','','','','','','C
ANCELLED')
[representation] add instance 75|Pat Porto|1116 Silverleaf
Dr|Arnold|MD|21012-1941|US|410-544-
2450||pfporto@msn.com|jmapat|||pporto#hotmail.com|||||APPROV
ED
```

```
[representation] size= 19
insert into 2PROFILE values (null,'75','Pat Porto','1116
Silverleaf Dr','Arnold','MD','21012-1941','US','410-544-
2450','','pfporto@msn.com','jmapat','','','pporto#hotmail.co
m','','','','','APPROVED')
[representation] add instance 76|Eric Lafontaine|429 de la
Montagne|Clericy|Quebec|J0Z1P0|Canada|819-637-
5536|None|lafbiz@yahoo.ca|||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values (null,'76','Eric
Lafontaine','429 de la
Montagne','Clericy','Quebec','J0Z1P0','Canada','819-637-
5536','None','lafbiz@yahoo.ca','','','','','','','','','APPR
OVED')
[representation] add instance 77|John Maille|128 woodridge
cres
#4|Nepean|Ontario|K2B7S9|Canada|||jjjok@rogers.com|||||||||A
PPROVED
[representation] size= 19
insert into 2PROFILE values (null,'77','John Maille','128
woodridge cres
#4','Nepean','Ontario','K2B7S9','Canada','','','jjjok@rogers
.com','','','','','','','','','','APPROVED')
[representation] add instance
78|Peter|Fehr|leamington|ONTARIO|N8H 5E9|Canada|519-324-
9542||fehrkp@sympatico.ca|||||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values
(null,'78','Peter','Fehr','leamington','ONTARIO','N8H
5E9','Canada','519-324-
9542','','fehrkp@sympatico.ca','','','','','','','','','CANC
ELLED')
[representation] add instance 79|JP Cooper|6432 Calle Vista
Dr.|El
Paso|TX|79912|USA|||jpc225@zwallet.com|||||||||CANCELLED
[representation] size= 19
insert into 2PROFILE values (null,'79','JP Cooper','6432
Calle Vista Dr.','El
Paso','TX','79912','USA','','','jpc225@zwallet.com','','','','
','','','','','','CANCELLED')
[representation] add instance 80|alex
burn|foredyke|hull|yorks|hu70dy|uk|||alexburn@homeworking.or
g|||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values (null,'80','alex
burn','foredyke','hull','yorks','hu70dy','uk','','','alexbur
n@homeworking.org','','','','','','','','','','APPROVED')
[representation] add instance 81|Dennis Renwick|55 Wayford
Crescent|Scarborough|Ontario|M1B 3E3|Canada|416-283-
0613|416-283-3137|socamaster@rogers.com|||||||||APPROVED
[representation] size= 19
insert into 2PROFILE values (null,'81','Dennis Renwick','55
Wayford Crescent','Scarborough','Ontario','M1B
3E3','Canada','416-283-0613','416-283-
```

```
3137','socamaster@rogers.com','','','','','','','','','APPRO
VED')
get instances from T3
select * from T3


[representation] clean tables
[attributename] new attribute name
[dbunifier] evaluate1: total attributes to eval = 30;
source1 = 1

[attributename] evaluate by attribute name
[attributename] analyze 5 x 25
[attributename] size of results array 125
[representation] store results attributename
[representation] inserting 0
[representation] inserting 1
[representation] inserting 2
[representation] inserting 3
[representation] inserting 4
[representation] inserting 5
[representation] inserting 6
[representation] inserting 7
[representation] inserting 8
[representation] inserting 9
[representation] inserting 10
[representation] inserting 11
[representation] inserting 12
[representation] inserting 13
[representation] inserting 14
[representation] inserting 15
[representation] inserting 16
[representation] inserting 17
[representation] inserting 18
[representation] inserting 19
[representation] inserting 20
[representation] inserting 21
[representation] inserting 22
[representation] inserting 23
[representation] inserting 24
[representation] inserting 25
[representation] inserting 26
[representation] inserting 27
[representation] inserting 28
[representation] inserting 29
[representation] inserting 30
[representation] inserting 31
[representation] inserting 32
[representation] inserting 33
[representation] inserting 34
[representation] inserting 35
[representation] inserting 36
[representation] inserting 37
[representation] inserting 38
```

```
[representation] inserting 39
[representation] inserting 40
[representation] inserting 41
[representation] inserting 42
[representation] inserting 43
[representation] inserting 44
[representation] inserting 45
[representation] inserting 46
[representation] inserting 47
[representation] inserting 48
[representation] inserting 49
[representation] inserting 50
[representation] inserting 51
[representation] inserting 52
[representation] inserting 53
[representation] inserting 54
[representation] inserting 55
[representation] inserting 56
[representation] inserting 57
[representation] inserting 58
[representation] inserting 59
[representation] inserting 60
[representation] inserting 61
[representation] inserting 62
[representation] inserting 63
[representation] inserting 64
[representation] inserting 65
[representation] inserting 66
[representation] inserting 67
[representation] inserting 68
[representation] inserting 69
[representation] inserting 70
[representation] inserting 71
[representation] inserting 72
[representation] inserting 73
[representation] inserting 74
[representation] inserting 75
[representation] inserting 76
[representation] inserting 77
[representation] inserting 78
[representation] inserting 79
[representation] inserting 80
[representation] inserting 81
[representation] inserting 82
[representation] inserting 83
[representation] inserting 84
[representation] inserting 85
[representation] inserting 86
[representation] inserting 87
[representation] inserting 88
[representation] inserting 89
[representation] inserting 90
[representation] inserting 91
[representation] inserting 92
```

```
[representation] inserting 93
[representation] inserting 94
[representation] inserting 95
[representation] inserting 96
[representation] inserting 97
[representation] inserting 98
[representation] inserting 99
[representation] inserting 100
[representation] inserting 101
[representation] inserting 102
[representation] inserting 103
[representation] inserting 104
[representation] inserting 105
[representation] inserting 106
[representation] inserting 107
[representation] inserting 108
[representation] inserting 109
[representation] inserting 110
[representation] inserting 111
[representation] inserting 112
[representation] inserting 113
[representation] inserting 114
[representation] inserting 115
[representation] inserting 116
[representation] inserting 117
[representation] inserting 118
[representation] inserting 119
[representation] inserting 120
[representation] inserting 121
[representation] inserting 122
[representation] inserting 123
[representation] inserting 124
[attributenamesubstring] new attribute name substring object
[dbunifier] evaluate1: total attributes to eval = 30;
source1 = 1

[attributenamesubstring] evaluate by attribute name
substrings
[attributenamesubstring] analyze 5 x 25
[attributenamesubstring] size of results array is 125
[representation] store results attributenamesubstring
[representation] inserting 0
[representation] inserting 1
[representation] inserting 2
[representation] inserting 3
[representation] inserting 4
[representation] inserting 5
[representation] inserting 6
[representation] inserting 7
[representation] inserting 8
[representation] inserting 9
[representation] inserting 10
[representation] inserting 11
[representation] inserting 12
```

```
[representation] inserting 13
[representation] inserting 14
[representation] inserting 15
[representation] inserting 16
[representation] inserting 17
[representation] inserting 18
[representation] inserting 19
[representation] inserting 20
[representation] inserting 21
[representation] inserting 22
[representation] inserting 23
[representation] inserting 24
[representation] inserting 25
[representation] inserting 26
[representation] inserting 27
[representation] inserting 28
[representation] inserting 29
[representation] inserting 30
[representation] inserting 31
[representation] inserting 32
[representation] inserting 33
[representation] inserting 34
[representation] inserting 35
[representation] inserting 36
[representation] inserting 37
[representation] inserting 38
[representation] inserting 39
[representation] inserting 40
[representation] inserting 41
[representation] inserting 42
[representation] inserting 43
[representation] inserting 44
[representation] inserting 45
[representation] inserting 46
[representation] inserting 47
[representation] inserting 48
[representation] inserting 49
[representation] inserting 50
[representation] inserting 51
[representation] inserting 52
[representation] inserting 53
[representation] inserting 54
[representation] inserting 55
[representation] inserting 56
[representation] inserting 57
[representation] inserting 58
[representation] inserting 59
[representation] inserting 60
[representation] inserting 61
[representation] inserting 62
[representation] inserting 63
[representation] inserting 64
[representation] inserting 65
[representation] inserting 66
```

```
[representation] inserting 67
[representation] inserting 68
[representation] inserting 69
[representation] inserting 70
[representation] inserting 71
[representation] inserting 72
[representation] inserting 73
[representation] inserting 74
[representation] inserting 75
[representation] inserting 76
[representation] inserting 77
[representation] inserting 78
[representation] inserting 79
[representation] inserting 80
[representation] inserting 81
[representation] inserting 82
[representation] inserting 83
[representation] inserting 84
[representation] inserting 85
[representation] inserting 86
[representation] inserting 87
[representation] inserting 88
[representation] inserting 89
[representation] inserting 90
[representation] inserting 91
[representation] inserting 92
[representation] inserting 93
[representation] inserting 94
[representation] inserting 95
[representation] inserting 96
[representation] inserting 97
[representation] inserting 98
[representation] inserting 99
[representation] inserting 100
[representation] inserting 101
[representation] inserting 102
[representation] inserting 103
[representation] inserting 104
[representation] inserting 105
[representation] inserting 106
[representation] inserting 107
[representation] inserting 108
[representation] inserting 109
[representation] inserting 110
[representation] inserting 111
[representation] inserting 112
[representation] inserting 113
[representation] inserting 114
[representation] inserting 115
[representation] inserting 116
[representation] inserting 117
[representation] inserting 118
[representation] inserting 119
[representation] inserting 120
```

```
[representation] inserting 121
[representation] inserting 122
[representation] inserting 123
[representation] inserting 124
[attributeinstance] new attribute instance object
[dbunifier] evaluate2: total attributes to eval = 30;
source1 = 1
[dbunifier] evaluate by attribute instances
[dbunifier] analyze 5 x 25
[representation] get instance 1 abstract35@hotmail.com
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[representation] get instance 1 johan_casper@yahoo.co.uk
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 2 1
[representation] get instance 2 55
[representation] get instance 2 56
[representation] get instance 2 57
[representation] get instance 2 58
[representation] get instance 2 59
[representation] get instance 2 60
[representation] get instance 2 61
[representation] get instance 2 62
[representation] get instance 2 63
[representation] get instance 2 64
[representation] get instance 2 65
[representation] get instance 2 69
[representation] get instance 2 70
[representation] get instance 2 71
[representation] get instance 2 72
[representation] get instance 2 73
[representation] get instance 2 74
[representation] get instance 2 75
[representation] get instance 2 76
[representation] get instance 2 77
[representation] get instance 2 78
[representation] get instance 2 79
[representation] get instance 2 80
[representation] get instance 2 81
[attributeinstance] final score = 0
```

```
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 alex burn
[representation] get instance 2 Art Hogenbirk
[representation] get instance 2 bill demarzo
[representation] get instance 2 Connie Blango
[representation] get instance 2 david cryer
[representation] get instance 2 Dennis Renwick
[representation] get instance 2 Dietmar Heinrich
[representation] get instance 2 Don
[representation] get instance 2 Eric Lafontaine
[representation] get instance 2 Ged Matthews
[representation] get instance 2 Heather Proud
[representation] get instance 2 John Maille
[representation] get instance 2 John Tanner
[representation] get instance 2 JP Cooper
[representation] get instance 2 Mike
[representation] get instance 2 Nathaniel
[representation] get instance 2 none
[representation] get instance 2 old username and password
[representation] get instance 2 Pat Porto
[representation] get instance 2 Peter
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 105 Nerepis Road
[representation] get instance 2 1116 Silverleaf Dr
[representation] get instance 2 128 woodridge cres #4
[representation] get instance 2 16 Three Hill View
[representation] get instance 2 19 orangeburg rd
[representation] get instance 2 312 Missouri Avenue, NW
[representation] get instance 2 3303 LaSalle
[representation] get instance 2 429 de la Montagne
[representation] get instance 2 55 Wayford Crescent
[representation] get instance 2 6315 solandra dr
[representation] get instance 2 6432 Calle Vista Dr.
[representation] get instance 2 Borova 536
[representation] get instance 2 Box 14335
[representation] get instance 2 Fehr
[representation] get instance 2 foredyke
[representation] get instance 2 Glendale, Satley
[representation] get instance 2 PO Box 1265
[representation] get instance 2 Walters
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Alton
[representation] get instance 2 Arnold
[representation] get instance 2 Bishop Auckland
[representation] get instance 2 Clericy
[representation] get instance 2 El Paso
[representation] get instance 2 Glastonbury
[representation] get instance 2 Grandbay/Westfield
[representation] get instance 2 hull
[representation] get instance 2 jacksonville
[representation] get instance 2 Kill Devil Hills
[representation] get instance 2 leamington
[representation] get instance 2 Miami
[representation] get instance 2 Nepean
[representation] get instance 2 old tappan
[representation] get instance 2 Pardubice
[representation] get instance 2 Scarborough
[representation] get instance 2 Spokane
[representation] get instance 2 Washington
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 County durham
[representation] get instance 2 DC
[representation] get instance 2 FL
[representation] get instance 2 fla
[representation] get instance 2 Illinois
[representation] get instance 2 MD
[representation] get instance 2 n/a
[representation] get instance 2 NB
[representation] get instance 2 NC
[representation] get instance 2 new jersey
[representation] get instance 2 Ontario
[representation] get instance 2 ONTARIO
[representation] get instance 2 Ontario
[representation] get instance 2 Quebec
[representation] get instance 2 Somerset
[representation] get instance 2 TX
[representation] get instance 2 WA
[representation] get instance 2 yorks
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 07675
[representation] get instance 2 20011
[representation] get instance 2 21012-1941
[representation] get instance 2 27948
[representation] get instance 2 32210
[representation] get instance 2 33162
[representation] get instance 2 53351
[representation] get instance 2 62002
[representation] get instance 2 79912
[representation] get instance 2 99214
[representation] get instance 2 BA6 8AU
[representation] get instance 2 DL13 4HT
[representation] get instance 2 E5K 2Z1
[representation] get instance 2 hu70dy
[representation] get instance 2 J0Z1P0
[representation] get instance 2 K2B7S9
[representation] get instance 2 M1B 3E3
[representation] get instance 2 N8H 5E9
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Czech Republic
[representation] get instance 2 U.S.A.
[representation] get instance 2 UK
[representation] get instance 2 uk
[representation] get instance 2 United kingdom
[representation] get instance 2 US
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 (202)882-1578
[representation] get instance 2 +440191 3504979
[representation] get instance 2 0042046653l337
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 2017675431
[representation] get instance 2 410-544-2450
[representation] get instance 2 416-283-0613
[representation] get instance 2 506 757 2246
[representation] get instance 2 519-324-9542
[representation] get instance 2 819-637-5536
[representation] get instance 2 904-7774981
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 416-283-3137
[representation] get instance 2 None
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 alexburn@homeworking.org
[representation] get instance 2 arthogenbirk@sprint.ca
[representation] get instance 2 bidderssweet@yahoo.com
[representation] get instance 2 cblango@aol.com
[representation] get instance 2 crye1217@msn.com
[representation] get instance 2 demarzob@bellatlantic.net
[representation] get instance 2 dietmar215@yahoo.de
[representation] get instance 2 eaglenet25@ureach.com
[representation] get instance 2 fehrkp@sympatico.ca
[representation] get instance 2 ged@choices4u.co.uk
[representation] get instance 2 jjjok@rogers.com
[representation] get instance 2 jpc225@zwallet.com
[representation] get instance 2 lafbiz@yahoo.ca
[representation] get instance 2 none@9000freeleads.net
[representation] get instance 2 nwalters1978@hotmail.com
[representation] get instance 2 oldaccount@9000freeleads.net
[representation] get instance 2 ozzyheather@yahoo.com
[representation] get instance 2 pfporto@msn.com
[representation] get instance 2 socamaster@rogers.com
[representation] get instance 2 tanman@ezl.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 jmapat
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 dcc272003
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 GednKev@choices4U.co.uk
[representation] get instance 2 nwalters589
[representation] get instance 2 pporto#hotmail.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 EXPIRED
[attributeinstance] final score = 0
```

```
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 1 http://adsaturation.com/cgi-
bin/sa/d.cgi/FL9450/links.html
[representation] get instance 1
http://www.adminder.com/c.cgiNULLabstract&worldsub
[representation] get instance 1
http://www.artzweb.net/domainname/domain_name_registration.p
hp
[representation] get instance 1
http://www.artzweb.net/webhosting/compare_plans.php
[representation] get instance 1
http://www.artzweb.net/webhosting/free_uk_dialup.php
[representation] get instance 1
http://www.artzweb.net/webhosting/small_business_hosting_ser
vice.php
[representation] get instance 1
http://www.artzweb.net/webhosting/starter_hosting_service.ph
p
[representation] get instance 1 http://www.distractacom.com
[representation] get instance 1 http://www.namesatnames.net
[representation] get instance 1
http://www.westhamfans.org/articles/transfer_talk.htm
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 2 1
[representation] get instance 2 55
[representation] get instance 2 56
[representation] get instance 2 57
[representation] get instance 2 58
[representation] get instance 2 59
[representation] get instance 2 60
[representation] get instance 2 61
[representation] get instance 2 62
[representation] get instance 2 63
[representation] get instance 2 64
[representation] get instance 2 65
[representation] get instance 2 69
[representation] get instance 2 70
[representation] get instance 2 71
[representation] get instance 2 72
[representation] get instance 2 73
[representation] get instance 2 74
[representation] get instance 2 75
[representation] get instance 2 76
[representation] get instance 2 77
```

```
[representation] get instance 2 78
[representation] get instance 2 79
[representation] get instance 2 80
[representation] get instance 2 81
[attributeinstance] final score = 0

[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 alex burn
[representation] get instance 2 Art Hogenbirk
[representation] get instance 2 bill demarzo
[representation] get instance 2 Connie Blango
[representation] get instance 2 david cryer
[representation] get instance 2 Dennis Renwick
[representation] get instance 2 Dietmar Heinrich
[representation] get instance 2 Don
[representation] get instance 2 Eric Lafontaine
[representation] get instance 2 Ged Matthews
[representation] get instance 2 Heather Proud
[representation] get instance 2 John Maille
[representation] get instance 2 John Tanner
[representation] get instance 2 JP Cooper
[representation] get instance 2 Mike
[representation] get instance 2 Nathaniel
[representation] get instance 2 none
[representation] get instance 2 old username and password
[representation] get instance 2 Pat Porto
[representation] get instance 2 Peter
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 105 Nerepis Road
[representation] get instance 2 1116 Silverleaf Dr
[representation] get instance 2 128 woodridge cres #4
[representation] get instance 2 16 Three Hill View
[representation] get instance 2 19 orangeburg rd
[representation] get instance 2 312 Missouri Avenue, NW
[representation] get instance 2 3303 LaSalle
[representation] get instance 2 429 de la Montagne
[representation] get instance 2 55 Wayford Crescent
[representation] get instance 2 6315 solandra dr
[representation] get instance 2 6432 Calle Vista Dr.
[representation] get instance 2 Borova 536
[representation] get instance 2 Box 14335
[representation] get instance 2 Fehr
```

```
[representation] get instance 2 foredyke
[representation] get instance 2 Glendale, Satley
[representation] get instance 2 PO Box 1265
[representation] get instance 2 Walters
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Alton
[representation] get instance 2 Arnold
[representation] get instance 2 Bishop Auckland
[representation] get instance 2 Clericy
[representation] get instance 2 El Paso
[representation] get instance 2 Glastonbury
[representation] get instance 2 Grandbay/Westfield
[representation] get instance 2 hull
[representation] get instance 2 jacksonville
[representation] get instance 2 Kill Devil Hills
[representation] get instance 2 leamington
[representation] get instance 2 Miami
[representation] get instance 2 Nepean
[representation] get instance 2 old tappan
[representation] get instance 2 Pardubice
[representation] get instance 2 Scarborough
[representation] get instance 2 Spokane
[representation] get instance 2 Washington
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 County durham
[representation] get instance 2 DC
[representation] get instance 2 FL
[representation] get instance 2 fla
[representation] get instance 2 Illinois
[representation] get instance 2 MD
[representation] get instance 2 n/a
[representation] get instance 2 NB
[representation] get instance 2 NC
[representation] get instance 2 new jersey
[representation] get instance 2 Ontario
[representation] get instance 2 ONTARIO
[representation] get instance 2 Ontario
[representation] get instance 2 Quebec
```

```
[representation] get instance 2 Somerset
[representation] get instance 2 TX
[representation] get instance 2 WA
[representation] get instance 2 yorks
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 07675
[representation] get instance 2 20011
[representation] get instance 2 21012-1941
[representation] get instance 2 27948
[representation] get instance 2 32210
[representation] get instance 2 33162
[representation] get instance 2 53351
[representation] get instance 2 62002
[representation] get instance 2 79912
[representation] get instance 2 99214
[representation] get instance 2 BA6 8AU
[representation] get instance 2 DL13 4HT
[representation] get instance 2 E5K 2Z1
[representation] get instance 2 hu70dy
[representation] get instance 2 J0Z1P0
[representation] get instance 2 K2B7S9
[representation] get instance 2 M1B 3E3
[representation] get instance 2 N8H 5E9
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Czech Republic
[representation] get instance 2 U.S.A.
[representation] get instance 2 UK
[representation] get instance 2 uk
[representation] get instance 2 United kingdom
[representation] get instance 2 US
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
```

```
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 (202)882-1578
[representation] get instance 2 +440191 3504979
[representation] get instance 2 00420466531337
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 2017675431
[representation] get instance 2 410-544-2450
[representation] get instance 2 416-283-0613
[representation] get instance 2 506 757 2246
[representation] get instance 2 519-324-9542
[representation] get instance 2 819-637-5536
[representation] get instance 2 904-7774981
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 416-283-3137
[representation] get instance 2 None
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 alexburn@homeworking.org
[representation] get instance 2 arthogenbirk@sprint.ca
[representation] get instance 2 bidderssweet@yahoo.com
[representation] get instance 2 cblango@aol.com
[representation] get instance 2 crye1217@msn.com
[representation] get instance 2 demarzob@bellatlantic.net
[representation] get instance 2 dietmar215@yahoo.de
[representation] get instance 2 eaglenet25@ureach.com
[representation] get instance 2 fehrkp@sympatico.ca
[representation] get instance 2 ged@choices4u.co.uk
[representation] get instance 2 jjjok@rogers.com
[representation] get instance 2 jpc225@zwallet.com
[representation] get instance 2 lafbiz@yahoo.ca
[representation] get instance 2 none@9000freeleads.net
[representation] get instance 2 nwalters1978@hotmail.com
[representation] get instance 2 oldaccount@9000freeleads.net
[representation] get instance 2 ozzyheather@yahoo.com
[representation] get instance 2 pfporto@msn.com
[representation] get instance 2 socamaster@rogers.com
[representation] get instance 2 tanman@ezl.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 jmapat
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 dcc272003
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 GednKev@choices4U.co.uk
[representation] get instance 2 nwalters589
[representation] get instance 2 pporto#hotmail.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
```

```
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 EXPIRED
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 1 Compare ARTzWeb.net web
hosting services
[representation] get instance 1 Distractacom network of
Internet sites.
[representation] get instance 1 Domain name registration
services. Register your domain name
[representation] get instance 1 Free ISP services to all UK
based customers
[representation] get instance 1 How to Generate Guaranteed
Traffic to Your Opportunity
[representation] get instance 1 namesatnames - Domain names
and web hosting services.
[representation] get instance 1 Small business web hosting
service.
[representation] get instance 1 Starter web hosting service
pack! Only 9.99 euros per month
[representation] get instance 1 Submit your site to 250 00
ffa sites free
[representation] get instance 1 West-ham transfer talk
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 2 1
[representation] get instance 2 55
[representation] get instance 2 56
[representation] get instance 2 57
[representation] get instance 2 58
[representation] get instance 2 59
[representation] get instance 2 60
[representation] get instance 2 61
[representation] get instance 2 62
[representation] get instance 2 63
[representation] get instance 2 64
[representation] get instance 2 65
[representation] get instance 2 69
[representation] get instance 2 70
[representation] get instance 2 71
[representation] get instance 2 72
[representation] get instance 2 73
```

```
[representation] get instance 2 74
[representation] get instance 2 75
[representation] get instance 2 76
[representation] get instance 2 77
[representation] get instance 2 78
[representation] get instance 2 79
[representation] get instance 2 80
[representation] get instance 2 81
[attributeinstance] final score = 0

[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 alex burn
[representation] get instance 2 Art Hogenbirk
[representation] get instance 2 bill demarzo
[representation] get instance 2 Connie Blango
[representation] get instance 2 david cryer
[representation] get instance 2 Dennis Renwick
[representation] get instance 2 Dietmar Heinrich
[representation] get instance 2 Don
[representation] get instance 2 Eric Lafontaine
[representation] get instance 2 Ged Matthews
[representation] get instance 2 Heather Proud
[representation] get instance 2 John Maille
[representation] get instance 2 John Tanner
[representation] get instance 2 JP Cooper
[representation] get instance 2 Mike
[representation] get instance 2 Nathaniel
[representation] get instance 2 none
[representation] get instance 2 old username and password
[representation] get instance 2 Pat Porto
[representation] get instance 2 Peter
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 105 Nerepis Road
[representation] get instance 2 1116 Silverleaf Dr
[representation] get instance 2 128 woodridge cres #4
[representation] get instance 2 16 Three Hill View
[representation] get instance 2 19 orangeburg rd
[representation] get instance 2 312 Missouri Avenue, NW
[representation] get instance 2 3303 LaSalle
[representation] get instance 2 429 de la Montagne
[representation] get instance 2 55 Wayford Crescent
[representation] get instance 2 6315 solandra dr
```

```
[representation] get instance 2 6432 Calle Vista Dr.
[representation] get instance 2 Borova 536
[representation] get instance 2 Box 14335
[representation] get instance 2 Fehr
[representation] get instance 2 foredyke
[representation] get instance 2 Glendale, Satley
[representation] get instance 2 PO Box 1265
[representation] get instance 2 Walters
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Alton
[representation] get instance 2 Arnold
[representation] get instance 2 Bishop Auckland
[representation] get instance 2 Clericy
[representation] get instance 2 El Paso
[representation] get instance 2 Glastonbury
[representation] get instance 2 Grandbay/Westfield
[representation] get instance 2 hull
[representation] get instance 2 jacksonville
[representation] get instance 2 Kill Devil Hills
[representation] get instance 2 leamington
[representation] get instance 2 Miami
[representation] get instance 2 Nepean
[representation] get instance 2 old tappan
[representation] get instance 2 Pardubice
[representation] get instance 2 Scarborough
[representation] get instance 2 Spokane
[representation] get instance 2 Washington
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 County durham
[representation] get instance 2 DC
[representation] get instance 2 FL
[representation] get instance 2 fla
[representation] get instance 2 Illinois
[representation] get instance 2 MD
[representation] get instance 2 n/a
[representation] get instance 2 NB
[representation] get instance 2 NC
[representation] get instance 2 new jersey
```

```
[representation] get instance 2 Ontario
[representation] get instance 2 ONTARIO
[representation] get instance 2 Ontario
[representation] get instance 2 Quebec
[representation] get instance 2 Somerset
[representation] get instance 2 TX
[representation] get instance 2 WA
[representation] get instance 2 yorks
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 07675
[representation] get instance 2 20011
[representation] get instance 2 21012-1941
[representation] get instance 2 27948
[representation] get instance 2 32210
[representation] get instance 2 33162
[representation] get instance 2 53351
[representation] get instance 2 62002
[representation] get instance 2 79912
[representation] get instance 2 99214
[representation] get instance 2 BA6 8AU
[representation] get instance 2 DL13 4HT
[representation] get instance 2 E5K 2Z1
[representation] get instance 2 hu70dy
[representation] get instance 2 J0Z1P0
[representation] get instance 2 K2B7S9
[representation] get instance 2 M1B 3E3
[representation] get instance 2 N8H 5E9
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Czech Republic
[representation] get instance 2 U.S.A.
[representation] get instance 2 UK
[representation] get instance 2 uk
[representation] get instance 2 United kingdom
```

```
[representation] get instance 2 US
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 (202)882-1578
[representation] get instance 2 +440191 3504979
[representation] get instance 2 00420466531337
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 2017675431
[representation] get instance 2 410-544-2450
[representation] get instance 2 416-283-0613
[representation] get instance 2 506 757 2246
[representation] get instance 2 519-324-9542
[representation] get instance 2 819-637-5536
[representation] get instance 2 904-7774981
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 416-283-3137
[representation] get instance 2 None
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 alexburn@homeworking.org
[representation] get instance 2 arthogenbirk@sprint.ca
[representation] get instance 2 bidderssweet@yahoo.com
[representation] get instance 2 cblango@aol.com
[representation] get instance 2 crye1217@msn.com
[representation] get instance 2 demarzob@bellatlantic.net
[representation] get instance 2 dietmar215@yahoo.de
[representation] get instance 2 eaglenet25@ureach.com
[representation] get instance 2 fehrkp@sympatico.ca
[representation] get instance 2 ged@choices4u.co.uk
[representation] get instance 2 jjjok@rogers.com
[representation] get instance 2 jpc225@zwallet.com
[representation] get instance 2 lafbiz@yahoo.ca
[representation] get instance 2 none@9000freeleads.net
[representation] get instance 2 nwalters1978@hotmail.com
[representation] get instance 2 oldaccount@9000freeleads.net
[representation] get instance 2 ozzyheather@yahoo.com
[representation] get instance 2 pfporto@msn.com
[representation] get instance 2 socamaster@rogers.com
[representation] get instance 2 tanman@ezl.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 jmapat
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 dcc272003
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 GednKev@choices4U.co.uk
[representation] get instance 2 nwalters589
[representation] get instance 2 pporto#hotmail.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
```

```
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 EXPIRED
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 1 1
[representation] get instance 1 1
[representation] get instance 1 1
[representation] get instance 1 2
[representation] get instance 1 2
[representation] get instance 1 2
[representation] get instance 1 2
[representation] get instance 1 2
[representation] get instance 1 2
[representation] get instance 1 6
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 2 1
[representation] get instance 2 55
[representation] get instance 2 56
[representation] get instance 2 57
[representation] get instance 2 58
[representation] get instance 2 59
[representation] get instance 2 60
[representation] get instance 2 61
[representation] get instance 2 62
[representation] get instance 2 63
[representation] get instance 2 64
[representation] get instance 2 65
[representation] get instance 2 69
[representation] get instance 2 70
[representation] get instance 2 71
[representation] get instance 2 72
[representation] get instance 2 73
[representation] get instance 2 74
[representation] get instance 2 75
[representation] get instance 2 76
[representation] get instance 2 77
[representation] get instance 2 78
```

```
[representation] get instance 2 79
[representation] get instance 2 80
[representation] get instance 2 81
[attributeinstance] final score = 0

[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 alex burn
[representation] get instance 2 Art Hogenbirk
[representation] get instance 2 bill demarzo
[representation] get instance 2 Connie Blango
[representation] get instance 2 david cryer
[representation] get instance 2 Dennis Renwick
[representation] get instance 2 Dietmar Heinrich
[representation] get instance 2 Don
[representation] get instance 2 Eric Lafontaine
[representation] get instance 2 Ged Matthews
[representation] get instance 2 Heather Proud
[representation] get instance 2 John Maille
[representation] get instance 2 John Tanner
[representation] get instance 2 JP Cooper
[representation] get instance 2 Mike
[representation] get instance 2 Nathaniel
[representation] get instance 2 none
[representation] get instance 2 old username and password
[representation] get instance 2 Pat Porto
[representation] get instance 2 Peter
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 105 Nerepis Road
[representation] get instance 2 1116 Silverleaf Dr
[representation] get instance 2 128 woodridge cres #4
[representation] get instance 2 16 Three Hill View
[representation] get instance 2 19 orangeburg rd
[representation] get instance 2 312 Missouri Avenue, NW
[representation] get instance 2 3303 LaSalle
[representation] get instance 2 429 de la Montagne
[representation] get instance 2 55 Wayford Crescent
[representation] get instance 2 6315 solandra dr
[representation] get instance 2 6432 Calle Vista Dr.
[representation] get instance 2 Borova 536
[representation] get instance 2 Box 14335
[representation] get instance 2 Fehr
[representation] get instance 2 foredyke
```

```
[representation] get instance 2 Glendale, Satley
[representation] get instance 2 PO Box 1265
[representation] get instance 2 Walters
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Alton
[representation] get instance 2 Arnold
[representation] get instance 2 Bishop Auckland
[representation] get instance 2 Clericy
[representation] get instance 2 El Paso
[representation] get instance 2 Glastonbury
[representation] get instance 2 Grandbay/Westfield
[representation] get instance 2 hull
[representation] get instance 2 jacksonville
[representation] get instance 2 Kill Devil Hills
[representation] get instance 2 leamington
[representation] get instance 2 Miami
[representation] get instance 2 Nepean
[representation] get instance 2 old tappan
[representation] get instance 2 Pardubice
[representation] get instance 2 Scarborough
[representation] get instance 2 Spokane
[representation] get instance 2 Washington
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 County durham
[representation] get instance 2 DC
[representation] get instance 2 FL
[representation] get instance 2 fla
[representation] get instance 2 Illinois
[representation] get instance 2 MD
[representation] get instance 2 n/a
[representation] get instance 2 NB
[representation] get instance 2 NC
[representation] get instance 2 new jersey
[representation] get instance 2 Ontario
[representation] get instance 2 ONTARIO
[representation] get instance 2 Ontario
[representation] get instance 2 Quebec
[representation] get instance 2 Somerset
```

```
[representation] get instance 2 TX
[representation] get instance 2 WA
[representation] get instance 2 yorks
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 07675
[representation] get instance 2 20011
[representation] get instance 2 21012-1941
[representation] get instance 2 27948
[representation] get instance 2 32210
[representation] get instance 2 33162
[representation] get instance 2 53351
[representation] get instance 2 62002
[representation] get instance 2 79912
[representation] get instance 2 99214
[representation] get instance 2 BA6 8AU
[representation] get instance 2 DL13 4HT
[representation] get instance 2 E5K 2Z1
[representation] get instance 2 hu70dy
[representation] get instance 2 J0Z1P0
[representation] get instance 2 K2B7S9
[representation] get instance 2 M1B 3E3
[representation] get instance 2 N8H 5E9
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Czech Republic
[representation] get instance 2 U.S.A.
[representation] get instance 2 UK
[representation] get instance 2 uk
[representation] get instance 2 United kingdom
[representation] get instance 2 US
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
```

```
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 (202)882-1578
[representation] get instance 2 +440191 3504979
[representation] get instance 2 00420466531337
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 2017675431
[representation] get instance 2 410-544-2450
[representation] get instance 2 416-283-0613
[representation] get instance 2 506 757 2246
[representation] get instance 2 519-324-9542
[representation] get instance 2 819-637-5536
[representation] get instance 2 904-7774981
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 416-283-3137
[representation] get instance 2 None
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 alexburn@homeworking.org
[representation] get instance 2 arthogenbirk@sprint.ca
[representation] get instance 2 bidderssweet@yahoo.com
[representation] get instance 2 cblango@aol.com
[representation] get instance 2 crye1217@msn.com
[representation] get instance 2 demarzob@bellatlantic.net
[representation] get instance 2 dietmar215@yahoo.de
[representation] get instance 2 eaglenet25@ureach.com
[representation] get instance 2 fehrkp@sympatico.ca
[representation] get instance 2 ged@choices4u.co.uk
[representation] get instance 2 jjjok@rogers.com
[representation] get instance 2 jpc225@zwallet.com
[representation] get instance 2 lafbiz@yahoo.ca
[representation] get instance 2 none@9000freeleads.net
[representation] get instance 2 nwalters1978@hotmail.com
[representation] get instance 2 oldaccount@9000freeleads.net
[representation] get instance 2 ozzyheather@yahoo.com
[representation] get instance 2 pfporto@msn.com
[representation] get instance 2 socamaster@rogers.com
[representation] get instance 2 tanman@ezl.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 jmapat
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 dcc272003
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 GednKev@choices4U.co.uk
[representation] get instance 2 nwalters589
[representation] get instance 2 pporto#hotmail.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
```

```
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 EXPIRED
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 1 20030218011051
[representation] get instance 1 20030218011354
[representation] get instance 1 20030218011650
[representation] get instance 1 20030218011951
[representation] get instance 1 20030218012253
[representation] get instance 1 20030218012331
[representation] get instance 1 20030218012556
[representation] get instance 1 20030218012903
[representation] get instance 1 20030218013201
[representation] get instance 1 20030218013454
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] get instance 2 1
[representation] get instance 2 55
[representation] get instance 2 56
[representation] get instance 2 57
[representation] get instance 2 58
[representation] get instance 2 59
[representation] get instance 2 60
[representation] get instance 2 61
[representation] get instance 2 62
[representation] get instance 2 63
[representation] get instance 2 64
[representation] get instance 2 65
[representation] get instance 2 69
[representation] get instance 2 70
[representation] get instance 2 71
[representation] get instance 2 72
[representation] get instance 2 73
[representation] get instance 2 74
[representation] get instance 2 75
[representation] get instance 2 76
[representation] get instance 2 77
[representation] get instance 2 78
[representation] get instance 2 79
[representation] get instance 2 80
[representation] get instance 2 81
[attributeinstance] final score = 0
```

```
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 admin
[representation] get instance 2 alex burn
[representation] get instance 2 Art Hogenbirk
[representation] get instance 2 bill demarzo
[representation] get instance 2 Connie Blango
[representation] get instance 2 david cryer
[representation] get instance 2 Dennis Renwick
[representation] get instance 2 Dietmar Heinrich
[representation] get instance 2 Don
[representation] get instance 2 Eric Lafontaine
[representation] get instance 2 Ged Matthews
[representation] get instance 2 Heather Proud
[representation] get instance 2 John Maille
[representation] get instance 2 John Tanner
[representation] get instance 2 JP Cooper
[representation] get instance 2 Mike
[representation] get instance 2 Nathaniel
[representation] get instance 2 none
[representation] get instance 2 old username and password
[representation] get instance 2 Pat Porto
[representation] get instance 2 Peter
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 105 Nerepis Road
[representation] get instance 2 1116 Silverleaf Dr
[representation] get instance 2 128 woodridge cres #4
[representation] get instance 2 16 Three Hill View
[representation] get instance 2 19 orangeburg rd
[representation] get instance 2 312 Missouri Avenue, NW
[representation] get instance 2 3303 LaSalle
[representation] get instance 2 429 de la Montagne
[representation] get instance 2 55 Wayford Crescent
[representation] get instance 2 6315 solandra dr
[representation] get instance 2 6432 Calle Vista Dr.
[representation] get instance 2 Borova 536
[representation] get instance 2 Box 14335
[representation] get instance 2 Fehr
[representation] get instance 2 foredyke
[representation] get instance 2 Glendale, Satley
[representation] get instance 2 PO Box 1265
[representation] get instance 2 Walters
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Alton
[representation] get instance 2 Arnold
[representation] get instance 2 Bishop Auckland
[representation] get instance 2 Clericy
[representation] get instance 2 El Paso
[representation] get instance 2 Glastonbury
[representation] get instance 2 Grandbay/Westfield
[representation] get instance 2 hull
[representation] get instance 2 jacksonville
[representation] get instance 2 Kill Devil Hills
[representation] get instance 2 leamington
[representation] get instance 2 Miami
[representation] get instance 2 Nepean
[representation] get instance 2 old tappan
[representation] get instance 2 Pardubice
[representation] get instance 2 Scarborough
[representation] get instance 2 Spokane
[representation] get instance 2 Washington
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 County durham
[representation] get instance 2 DC
[representation] get instance 2 FL
[representation] get instance 2 fla
[representation] get instance 2 Illinois
[representation] get instance 2 MD
[representation] get instance 2 n/a
[representation] get instance 2 NB
[representation] get instance 2 NC
[representation] get instance 2 new jersey
[representation] get instance 2 Ontario
[representation] get instance 2 ONTARIO
[representation] get instance 2 Ontario
[representation] get instance 2 Quebec
[representation] get instance 2 Somerset
[representation] get instance 2 TX
[representation] get instance 2 WA
[representation] get instance 2 yorks
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 07675
[representation] get instance 2 20011
[representation] get instance 2 21012-1941
[representation] get instance 2 27948
[representation] get instance 2 32210
[representation] get instance 2 33162
[representation] get instance 2 53351
[representation] get instance 2 62002
[representation] get instance 2 79912
[representation] get instance 2 99214
[representation] get instance 2 BA6 8AU
[representation] get instance 2 DL13 4HT
[representation] get instance 2 E5K 2Z1
[representation] get instance 2 hu70dy
[representation] get instance 2 J0Z1P0
[representation] get instance 2 K2B7S9
[representation] get instance 2 M1B 3E3
[representation] get instance 2 N8H 5E9
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Canada
[representation] get instance 2 Czech Republic
[representation] get instance 2 U.S.A.
[representation] get instance 2 UK
[representation] get instance 2 uk
[representation] get instance 2 United kingdom
[representation] get instance 2 US
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[representation] get instance 2 usa
[representation] get instance 2 USA
[representation] get instance 2 USA
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 (202)882-1578
[representation] get instance 2 +440191 3504979
[representation] get instance 2 00420466531337
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 2017675431
[representation] get instance 2 410-544-2450
[representation] get instance 2 416-283-0613
[representation] get instance 2 506 757 2246
[representation] get instance 2 519-324-9542
[representation] get instance 2 819-637-5536
[representation] get instance 2 904-7774981
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 1-877-843-4609
[representation] get instance 2 416-283-3137
[representation] get instance 2 None
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 alexburn@homeworking.org
[representation] get instance 2 arthogenbirk@sprint.ca
[representation] get instance 2 bidderssweet@yahoo.com
[representation] get instance 2 cblango@aol.com
[representation] get instance 2 crye1217@msn.com
[representation] get instance 2 demarzob@bellatlantic.net
[representation] get instance 2 dietmar215@yahoo.de
[representation] get instance 2 eaglenet25@ureach.com
[representation] get instance 2 fehrkp@sympatico.ca
[representation] get instance 2 ged@choices4u.co.uk
[representation] get instance 2 jjjok@rogers.com
[representation] get instance 2 jpc225@zwallet.com
[representation] get instance 2 lafbiz@yahoo.ca
[representation] get instance 2 none@9000freeleads.net
[representation] get instance 2 nwalters1978@hotmail.com
[representation] get instance 2 oldaccount@9000freeleads.net
[representation] get instance 2 ozzyheather@yahoo.com
[representation] get instance 2 pfporto@msn.com
[representation] get instance 2 socamaster@rogers.com
[representation] get instance 2 tanman@ezl.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 jmapat
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 dcc272003
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2 GednKev@choices4U.co.uk
[representation] get instance 2 nwalters589
[representation] get instance 2 pporto#hotmail.com
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0
```

```
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[representation] get instance 2
[attributeinstance] final score = 0

[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 APPROVED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 CANCELLED
[representation] get instance 2 EXPIRED
[attributeinstance] final score = 0
```

```
[attributeinstance] final score = 0

[attributeinstance] final score = 0

[representation] store results attributeinstance
[representation] inserting 0
[representation] inserting 1
[representation] inserting 2
[representation] inserting 3
[representation] inserting 4
[representation] inserting 5
[representation] inserting 6
[representation] inserting 7
[representation] inserting 8
[representation] inserting 9
[representation] inserting 10
[representation] inserting 11
[representation] inserting 12
[representation] inserting 13
[representation] inserting 14
[representation] inserting 15
[representation] inserting 16
[representation] inserting 17
[representation] inserting 18
[representation] inserting 19
[representation] inserting 20
[representation] inserting 21
[representation] inserting 22
[representation] inserting 23
[representation] inserting 24
[representation] inserting 25
[representation] inserting 26
[representation] inserting 27
[representation] inserting 28
[representation] inserting 29
[representation] inserting 30
[representation] inserting 31
[representation] inserting 32
[representation] inserting 33
[representation] inserting 34
[representation] inserting 35
[representation] inserting 36
[representation] inserting 37
[representation] inserting 38
[representation] inserting 39
[representation] inserting 40
[representation] inserting 41
[representation] inserting 42
[representation] inserting 43
[representation] inserting 44
[representation] inserting 45
[representation] inserting 46
[representation] inserting 47
[representation] inserting 48
```

```
[representation] inserting 49
[representation] inserting 50
[representation] inserting 51
[representation] inserting 52
[representation] inserting 53
[representation] inserting 54
[representation] inserting 55
[representation] inserting 56
[representation] inserting 57
[representation] inserting 58
[representation] inserting 59
[representation] inserting 60
[representation] inserting 61
[representation] inserting 62
[representation] inserting 63
[representation] inserting 64
[representation] inserting 65
[representation] inserting 66
[representation] inserting 67
[representation] inserting 68
[representation] inserting 69
[representation] inserting 70
[representation] inserting 71
[representation] inserting 72
[representation] inserting 73
[representation] inserting 74
[representation] inserting 75
[representation] inserting 76
[representation] inserting 77
[representation] inserting 78
[representation] inserting 79
[representation] inserting 80
[representation] inserting 81
[representation] inserting 82
[representation] inserting 83
[representation] inserting 84
[representation] inserting 85
[representation] inserting 86
[representation] inserting 87
[representation] inserting 88
[representation] inserting 89
[representation] inserting 90
[representation] inserting 91
[representation] inserting 92
[representation] inserting 93
[representation] inserting 94
[representation] inserting 95
[representation] inserting 96
[representation] inserting 97
[representation] inserting 98
[representation] inserting 99
[representation] inserting 100
[representation] inserting 101
[representation] inserting 102
```

```
[representation] inserting 103
[representation] inserting 104
[representation] inserting 105
[representation] inserting 106
[representation] inserting 107
[representation] inserting 108
[representation] inserting 109
[representation] inserting 110
[representation] inserting 111
[representation] inserting 112
[representation] inserting 113
[representation] inserting 114
[representation] inserting 115
[representation] inserting 116
[representation] inserting 117
[representation] inserting 118
[representation] inserting 119
[representation] inserting 120
[representation] inserting 121
[representation] inserting 122
[representation] inserting 123
[representation] inserting 124
```

APPENDIX C

Network Connection Statistics

```
Pinging marketingbots.com [64.191.25.145] with 32 bytes of
data:
Reply from 64.191.25.145: bytes=32 time=44ms TTL=243
Reply from 64.191.25.145: bytes=32 time=45ms TTL=243
Reply from 64.191.25.145: bytes=32 time=44ms TTL=243
Reply from 64.191.25.145: bytes=32 time=44ms TTL=243

Ping statistics for 64.191.25.145:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 44ms, Maximum =  45ms, Average =  44ms

 Tracing route to marketingbots.com [64.191.25.145] over a
maximum of 30 hops:
1     1 ms    <10 ms    <10 ms  192.168.1.254
2    <10 ms     1 ms      1 ms  gw.cocse.unf.edu [139.62.32.1]
3     2 ms      1 ms      1 ms  egret.noc.unf.edu
[139.62.228.1]
4     3 ms      3 ms      3 ms  172.17.82.105
5     3 ms      2 ms      4 ms  205.152.187.248
6    11 ms     11 ms     12 ms  axr00mia-7-0-0-0.bellsouth.net
[65.83.237.92]
7    10 ms     10 ms     11 ms  pxr00mia-2-0-0.bellsouth.net
[65.83.236.18]
8    12 ms     10 ms     11 ms  POS1-
0.hsipaccess2.Miami1.Level3.net [64.156.8.21]
9    11 ms     12 ms     12 ms  ge-6-1-1.mp2.Miami1.level3.net
[64.159.1.101]
10   40 ms     41 ms     40 ms  so-0-1-
0.mp2.Philadelphia1.Level3.net [64.159.0.142]
11   40 ms     41 ms     41 ms  gig10-
1.hsa1.Philadelphia1.level3.net [64.159.3.30]
12   47 ms     48 ms     49 ms  unknown.Level3.net
[64.157.236.2]
13   45 ms     47 ms     45 ms  marketingbots.com
[64.191.25.145]
Trace complete.
```

APPENDIX D

Conversions Needed between Oracle Meta-Data and MySQL Data Types

	Oracle	MySQL
0	# TABLE NAME	table
1	# COLUMN NAME	attribute
2	# DATA TYPE	type(size)
3	# DATA TYPE MOD	
4	# DATA TYPE OWNER	
5	# DATA LENGTH	type(size)
6	# DATA PRECISION	
7	# DATA SCALE	
8	# NULLABLE	null
9	# COLUMN ID	
10	# DEFAULT LENGTH	
11	# DATA DEFAULT	default
12	# NUM DISTINCT	
13	# LOW VALUE	
14	# HIGH VALUE	
15	# DENSITY	
16	# NUM NULLS	
17	# NUM BUCKETS	
18	# LAST ANALYZED	
19	# SAMPLE SIZE	
20	# CHARACTER SET NAME	
21	# CHAR COL DECL LENGTH	
22	# GLOBAL STATS	
23	# USER STATS	
24	# AVG COL LEN	

Table 8: Conversions Needed between Oracle Meta-Data and MySQL Data Types

APPENDIX E

Programs Used to Conduct Pre-Analysis

```
#!/usr/bin/perl

#############################################################
# program: Test access and selection types
#
# section: Chapter 4 - Pre-Analysis
#
# description: Simulates any program with large amounts of
data
# to be retrieved from a database.  Loads data sets of
# various sizes into a database and retrieves the data using
# several combinations of access methods.  A comparison of
# two items selected from the database is conducted.
#############################################################

# load required modules used to access the db
# and perform time tests
use DBI;
use Time::HiRes qw( usleep ualarm gettimeofday tv_interval
);

# clear the output file if one exists
open(FILE,">results.txt");
print FILE "- begin -\n";
close(FILE);

# these are the data set sizes representing the number of
# comparisons that are being conducted for the analysis
@amounts =
(500,1000,1500,2000,2500,3000,4000,5000,10000,15000,20000);

# clear contents of table being used to store the data sets
my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
$sql = "delete from test1";
$sth = $dbh->prepare($sql);
$sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
$dbh->disconnect;

# open connection to be used to access the database
```

```
my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});

@file="";
# obtain file containing the datasets being used
open(FILE,"scripts.txt");
for($i=0;$i<20000;$i++){
     push(@file,$_);
}
close(FILE);

# flip through the datasets and
# select the two items to be evaluated
foreach $line (@file){
     @each=split(/\,/, $line);
     push(@email,$each[4]);
     push(@username,$each[9]);
}

# write the results of the run to a text file for
# future use
open(FILE,">>results.txt");

# perform the analysis 50 times
for($a=0;$a<50;$a++){

     # perform a loop for each of the dataset sizes being
     # evaluated
     foreach $amount (@amounts){

          # signal that a next test has begun
          print FILE "\n\n- begin $amount -\n";
          print "\n\n- begin $amount -\n";

          # measure raw time doing comparison while in local
memory
          ($seconds1, $microseconds1) = gettimeofday;

          # perform the entire set of evaluations while
items
          # are in local memory
          for($i=0;$i<$amount;$i++){
               if($email[$i] =~ $username[$i]){
                    $result[$i]=1;
               }else{
                    $result[$i]=0;
               }
          }

          # measure raw time
          ($seconds2, $microseconds2) = gettimeofday;
```

```
          # insert all the dataset being evaluated into the
database
          for($i=0;$i<$amount;$i++){
               $sql = "insert into test1 values
('$email[$i]','$username[$i]',$result[$i])";
               $sth = $dbh->prepare($sql);
               $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
          }

          # measure raw time
          ($seconds3, $microseconds3) = gettimeofday;

          # select each one and do comparison
          for($i=0;$i<$amount;$i++){
               $sql = "select * from test1 where email =
'$email[$i]' limit 1";
               $sth = $dbh->prepare($sql);
               $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
               @instance = $sth->fetchrow;

               if($instance[0] =~ $instance[1]){
                    $result=1;
               }else{
                    $result=0;
               }
          }

          # measure raw time
          ($seconds4, $microseconds4) = gettimeofday;

          # select all of them and do comparison
          $sql = "select * from test1";
          $sth = $dbh->prepare($sql);
          $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
          while(@instance = $sth->fetchrow){
               if($instance[0] =~ $instance[1]){
                    $result=1;
               }else{
                    $result=0;
               }
          }

          # measure raw time
          ($seconds5, $microseconds5) = gettimeofday;

          # select each of them and get the result from the
database
          for($i=0;$i<$amount;$i++){
               $sql = "select * from test1 where email =
'$email[$i]' limit 1";
               $sth = $dbh->prepare($sql);
```

```
                $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
                @instance = $sth->fetchrow;
          }

          # measure raw time
          ($seconds6, $microseconds6) = gettimeofday;

          # select all of them and get the result from the
database
          $sql = "select * from test1";
          $sth = $dbh->prepare($sql);
          $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
          while(@instance = $sth->fetchrow){
                # result is now in memory as $instance[2]
                # items being compared are now in memory as
$instance[0]
                #         and $instance[1]
          }

          # measure raw time
          ($seconds7, $microseconds7) = gettimeofday;

          # don't measure this
          $dbh->disconnect;

          # measure raw time
          ($seconds8, $microseconds8) = gettimeofday;

##############################################################
###############

          # select each one and do comparison
          for($i=0;$i<$amount;$i++){
                my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
                $sql = "select * from test1 where email =
'$email[$i]' limit 1";
                $sth = $dbh->prepare($sql);
                $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
                @instance = $sth->fetchrow;

                if($instance[0] =~ $instance[1]){
                      $result=1;
                }else{
                      $result=0;
                }
                $sth->finish();
                $dbh->disconnect;
          }
```

```
          # measure raw time
          ($seconds9, $microseconds9) = gettimeofday;

          # select all of them and do comparison
          my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
          $sql = "select * from test1";
          $sth = $dbh->prepare($sql);
          $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
          while(@instance = $sth->fetchrow){
               if($instance[0] =~ $instance[1]){
                    $result=1;
               }else{
                    $result=0;
               }
          }
          $dbh->disconnect;

          # measure raw time
          ($seconds10, $microseconds10) = gettimeofday;

          # select each of them and get the result from the
database
          for($i=0;$i<$amount;$i++){
               my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
               $sql = "select * from test1 where email =
'$email[$i]' limit 1";
               $sth = $dbh->prepare($sql);
               $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
               @instance = $sth->fetchrow;
               $sth->finish();
               $dbh->disconnect;
          }

          # measure raw time
          ($seconds11, $microseconds11) = gettimeofday;

          # select all of them and get the result from the
database
          my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
          $sql = "select * from test1";
          $sth = $dbh->prepare($sql);
          $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
          while(@instance = $sth->fetchrow){

          }
```

```
            $dbh->disconnect;

            # measure raw time
            ($seconds12, $microseconds12) = gettimeofday;

            # evaluate time used to perform each of the
different
            # methods
            $dif1 = $seconds2-$seconds1+(($microseconds2-
$microseconds1)/1000000);
            $dif2 = $seconds3-$seconds2+(($microseconds3-
$microseconds2)/1000000);
            $dif3 = $seconds4-$seconds3+(($microseconds4-
$microseconds3)/1000000);
            $dif4 = $seconds5-$seconds4+(($microseconds5-
$microseconds4)/1000000);
            $dif5 = $seconds6-$seconds5+(($microseconds6-
$microseconds5)/1000000);
            $dif6 = $seconds7-$seconds6+(($microseconds7-
$microseconds6)/1000000);

            $difa = $seconds9-$seconds8+(($microseconds9-
$microseconds8)/1000000);
            $difb = $seconds10-$seconds9+(($microseconds10-
$microseconds9)/1000000);
            $difc = $seconds11-$seconds10+(($microseconds11-
$microseconds10)/1000000);
            $difd = $seconds12-$seconds11+(($microseconds12-
$microseconds11)/1000000);

            # display and store the time results from the run
            print "\nrun $a for $amount\n";
            print "difference 1 - $dif1<br>\n";
            print "difference 2 - $dif2<br>\n";
            print "difference 3 - $dif3<br>\n";
            print "difference 4 - $dif4<br>\n";
            print "difference 5 - $dif5<br>\n";
            print "difference 6 - $dif6<br>\n";
            print "difference a - $difa<br>\n";
            print "difference b - $difb<br>\n";
            print "difference c - $difc<br>\n";
            print "difference d - $difd<br>\n";

            print FILE
"$dif1|$dif2|$dif3|$dif4|$dif5|$dif6|$difa|$difb|$difc|$difd
\n";

            # clear the table being used to conduct the test
and disconnect
            my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
            $sql = "delete from test1";
            $sth = $dbh->prepare($sql);
```

```
            $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
            $dbh->disconnect;
      }
}
end;
```

```
#!/usr/bin/perl

#############################################################
# program: Analyses time required to see if an item is
#  already in the database
#
# section: Chapter 4 - Pre-Analysis
#
# description: Loads data sets of various sizes into
#  a database and then analyses the time required to
#  check if the items are in the database.
#############################################################

# load required modules used to access the db
# and perform time tests
use DBI;
use Time::HiRes qw( usleep ualarm gettimeofday tv_interval
);

# clear the output file if one exists
open(FILE,">results.txt");
print FILE "- begin -\n";
close(FILE);

# these are the data set sizes representing the number of
# comparisons that are being conducted for the analysis
@amounts =
(500,1000,1500,2000,2500,3000,4000,5000,10000,15000,20000);

# clear contents of table being used to store the data sets
my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});
$sql = "delete from test1";
$sth = $dbh->prepare($sql);
$sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
$dbh->disconnect;

# open connection to be used to access the database
my $dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker",{'RaiseError' => 1});

@file="";
# obtain file containing the datasets being used
open(FILE,"scripts.txt");
for($i=0;$i<20000;$i++){
     push(@file,$_);
}
close(FILE);

# flip through the datasets and
# select the two items to be evaluated
```

```
foreach $line (@file){
     @each=split(/\,/, $line);
     push(@email,$each[4]);
     push(@username,$each[9]);
}


for($i=0;$i<20000;$i++){
     if($email[$i] =~ $username[$i]){
          $result[$i]=1;
     }else{
          $result[$i]=0;
     }
}

# Load database with the test data
for($i=0;$i<20000;$i++){
     $sql = "insert into test1 values
('$email[$i]','$username[$i]',$result[$i])";
     $sth = $dbh->prepare($sql);
     $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
}

# write the results of the run to a text file for
# future use
open(FILE,">>results.txt");
foreach $amount (@amounts){
     print FILE "\n\n- begin $amount -\n";
     print "\n\n- begin $amount -\n";

     # perform the analysis 50 times
     for($a=0;$a<50;$a++){
          # measure raw time
          ($seconds3, $microseconds3) = gettimeofday;

          # perform a loop for each of the dataset sizes
being
          # evaluated
          for($i=0;$i<$amount;$i++){
               $sql = "select result from test1 where email
= '$email[$i]' and username = '$username[$i]' limit 1";
               $sth = $dbh->prepare($sql);
               $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
               @instance = $sth->fetchrow;

               # check to see if there are a data item
returned from
               # the database or not
               if($instance[0] == ""){
                    $result=1;
               }else{
                    $result=0;
```

```
                }
            }

            # measure raw time
            ($seconds4, $microseconds4) = gettimeofday;

            # calculate time needed to evaluate if the item
was in the database
            $dif3 = $seconds4-$seconds3+(($microseconds4-
$microseconds3)/1000000);

            # report the findings
            print "time measurement number $a - $dif3<br>\n";
            print FILE "$dif3\n";
        }
}
end;
```

APPENDIX F

Sample Test Data used for Pre-Analysis

A complete copy of the test data is available on the CD with this document.

Subscription Cancellation,3/12/2003,22:01:44,John Alderman,amerijohn@hotmail.com,,Active - Canceled - By Buyer,0,66.119.33.135,jevon,3/12/2003,22:01:44,

Subscription Cancellation,3/12/2003,21:59:54,John Alderman,amerijohn@hotmail.com,,Active - Canceled - By Buyer,0,216.148.244.38,jevon,3/12/2003,21:59:54,

Subscription Creation,3/12/2003,21:53:15,Susan Saunders,SWhiteDragon@aol.com,,Active,0,152.163.189.103,hekela76,,,

Subscription Creation,3/12/2003,21:07:47,John Alderman,amerijohn@hotmail.com,,Active - Canceled - By Buyer,0,216.148.244.38,jevon,3/12/2003,21:59:54,

Subscription Creation,3/12/2003,21:00:36,John Alderman,amerijohn@hotmail.com,,Active - Canceled - By Buyer,0,66.119.33.135,jevon,3/12/2003,22:01:44,

Subscription Creation,3/12/2003,20:20:44,scott bruce,thebruce42@msn.com,,Active,0,68.61.173.83,hekela76,,,

Subscription Creation,3/12/2003,19:23:23,Kenneth Hilton Saint,khsforu@aol.com,,Active,0,203.87.121.212,moneyaddict,,,

Subscription Creation,3/12/2003,18:31:37,Brien Bennett,brienbennett@xtra.co.nz,,Active,0,219.88.218.118,redsky,,,

Subscription Creation,3/12/2003,17:41:27,CAMILLE MULLINS,cammllns@aol.com,,Active,0,152.163.189.103,hekela76,,,

Subscription Cancellation,3/12/2003,17:40:57,Janice L Phillips,jl_phillips0525@msn.com,,Active - Canceled - By Buyer,0,66.119.34.39,wlwjsa,3/12/2003,17:40:57,

APPENDIX G

Pre-Analysis Results

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
500	1.21	0	1.81	0.16	1.76	0.11	7.86	0.22	7.69	0.16
500	1.1	0	1.75	0.17	1.76	0.16	7.74	0.17	7.74	0.16
500	1.1	0	1.75	0.17	1.76	0.16	7.74	0.17	7.69	0.16
500	1.1	0	1.76	0.16	1.71	0.16	7.74	0.22	7.75	0.16
500	1.09	0	1.76	0.17	1.75	0.17	7.74	0.16	7.74	0.17
500	1.04	0	1.76	0.17	1.75	0.17	7.74	0.16	7.74	0.17
500	1.1	0	1.76	0.16	1.76	0.11	7.74	0.22	7.75	0.17
500	1.05	0	1.75	0.22	1.71	0.16	7.74	0.17	7.74	0.16
500	1.1	0	1.76	0.16	1.76	0.16	7.74	0.17	7.75	0.16
500	1.1	0	1.76	0.16	1.76	0.16	7.75	0.17	7.75	0.16
500	1.04	0	1.76	0.16	1.76	0.17	7.74	0.16	7.74	0.17
500	1.1	0	1.76	0.16	1.76	0.11	7.75	0.22	7.74	0.16
500	1.04	0	1.76	0.17	1.75	0.17	7.74	0.16	7.74	0.17
500	1.09	0	1.76	0.17	1.75	0.17	7.75	0.16	7.75	0.17
500	1.1	0	1.76	0.17	1.75	0.17	7.74	0.16	7.74	0.17
500	1.04	0.06	1.76	0.16	1.76	0.16	7.75	0.17	7.74	0.16
500	1.04	0.06	1.76	0.16	1.76	0.17	7.69	0.16	7.75	0.17
500	1.1	0	1.75	0.17	1.7	0.17	7.74	0.22	7.74	0.16
500	1.1	0	1.76	0.16	1.71	0.16	7.75	0.17	7.74	0.22
500	1.1	0	1.81	0.16	1.76	0.17	7.75	0.16	7.74	0.17
500	1.1	0	1.76	0.16	1.76	0.17	7.75	0.16	7.74	0.17
500	1.1	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.17	1.7	0.16	0	0.22	0	0.17
500	1.1	0	1.76	0.16	1.71	0.16	0	0.22	0	0.17
500	1.09	0	1.76	0.17	1.7	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.16	1.71	0.16	0	0.22	0	0.17
500	1.1	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.05	0	1.75	0.17	1.76	0.16	0	0.17	0	0.22
500	1.1	0	1.76	0.16	1.76	0.17	0	0.16	0	0.17

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
500	1.1	0	1.76	0.16	1.76	0.17	0	0.16	0	0.17
500	1.1	0	1.75	0.17	1.76	0.16	0	0.16	0	0.17
500	1.1	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.05	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.16	1.76	0.17	0	0.16	0	0.17
500	1.05	0	1.75	0.17	1.76	0.16	0	0.16	0	0.17
500	1.1	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.04	0	1.82	0.16	1.76	0.11	0	0.22	0	0.16
500	1.05	0	1.75	0.17	1.76	0.16	0	0.16	0	0.22
500	1.05	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.05	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.05	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.16	1.76	0.16	0	0.17	0	0.16
500	1.04	0	1.76	0.16	1.76	0.17	0	0.16	0	0.16
500	1.04	0	1.76	0.17	1.75	0.17	0	0.16	0	0.17
500	1.1	0	1.75	0.17	1.76	0.16	0	0.17	0	0.16
500	1.1	0	1.76	0.16	1.71	0.16	0	0.22	0	0.17
500	1.1	0	1.76	0.17	1.75	0.17	0	0.16	0	0.17
Avg.	1.084	0.0024	1.7602	0.1658	1.749	0.1592	7.746667	0.1764	7.738095	0.1678

Table 9: Time Evaluations When 500 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
1000	2.2	0.06	3.84	0.33	3.68	0.33	15.6	0.33	15.66	0.33
1000	2.19	0	3.68	0.33	3.68	0.33	15.71	0.33	15.65	0.33
1000	2.19	0	3.68	0.33	3.68	0.33	15.65	0.33	15.71	0.33
1000	2.2	0	3.68	0.33	3.68	0.27	15.71	0.39	15.66	0.33
1000	2.2	0	3.68	0.33	3.68	0.33	15.65	0.33	15.71	0.33
1000	2.15	0	3.68	0.33	3.68	0.32	15.71	0.33	15.65	0.33
1000	2.19	0	3.68	0.33	3.63	0.33	15.71	0.33	15.65	0.33
1000	2.19	0	3.68	0.33	3.68	0.33	15.7	0.33	15.71	0.33
1000	2.14	0	3.73	0.33	3.68	0.33	15.71	0.33	15.66	0.33
1000	2.14	0	3.74	0.32	3.68	0.33	15.71	0.33	15.71	0.33
1000	2.2	0	3.68	0.33	3.68	0.33	15.76	0.33	15.65	0.33
1000	2.2	0	3.68	0.33	3.68	0.33	15.65	0.33	15.65	0.33
1000	2.15	0	3.68	0.32	3.68	0.33	15.65	0.33	15.66	0.33
1000	2.15	0	3.68	0.32	3.68	0.33	15.71	0.33	15.71	0.33
1000	2.2	0	3.68	0.33	3.68	0.33	15.71	0.33	15.71	0.33
1000	2.14	0	3.74	0.32	3.68	0.33	15.65	0.33	15.65	0.33
1000	2.14	0	3.74	0.33	3.68	0.33	15.66	0.33	15.65	0.33
1000	2.2	0	3.68	0.33	3.62	0.33	15.66	0.33	15.66	0.33
1000	2.19	0	3.68	0.33	3.68	0.33	15.65	0.33	15.66	0.33
1000	2.19	0	3.68	0.33	3.74	0.27	15.71	0.33	15.65	0.27
1000	2.2	0	3.68	0.33	3.62	0.33	15.71	0.33	15.71	0.33
1000	2.14	0.06	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.19	0	3.68	0.33	3.74	0.27	0	0.33	0	0.33
1000	2.19	0	3.68	0.33	3.63	0.33	0	0.33	0	0.33
1000	2.14	0	3.68	0.38	3.63	0.33	0	0.33	0	0.33
1000	2.14	0	3.73	0.33	3.74	0.27	0	0.39	0	0.33
1000	2.14	0	3.74	0.32	3.68	0.28	0	0.38	0	0.33
1000	2.19	0	3.68	0.33	3.63	0.33	0	0.33	0	0.33
1000	2.14	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.14	0.05	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.15	0.05	3.68	0.33	3.68	0.33	0	0.33	0	0.32
1000	2.14	0.06	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.14	0	3.73	0.33	3.63	0.33	0	0.33	0	0.33
1000	2.14	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.19	0	3.68	0.33	3.74	0.27	0	0.39	0	0.27
1000	2.15	0	3.73	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.2	0	3.68	0.33	3.68	0.27	0	0.39	0	0.33
1000	2.14	0	3.73	0.33	3.68	0.33	0	0.33	0	0.33

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
1000	2.2	0	3.68	0.39	3.68	0.32	0	0.33	0	0.33
1000	2.2	0	3.68	0.33	3.63	0.33	0	0.33	0	0.32
1000	2.14	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.14	0	3.68	0.33	3.68	0.33	0	0.33	0	0.38
1000	2.14	0	3.73	0.33	3.63	0.33	0	0.33	0	0.33
1000	2.14	0	3.73	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.2	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.14	0	3.74	0.33	3.62	0.33	0	0.33	0	0.33
1000	2.15	0	3.68	0.33	3.68	0.33	0	0.32	0	0.33
1000	2.15	0	3.73	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.19	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
1000	2.14	0	3.68	0.33	3.68	0.33	0	0.33	0	0.33
Avg.	2.1668	0.0056	3.6972	0.3312	3.6742	0.3214	15.68476	0.3356	15.67286	0.3282

Table 10: Time Evaluations When 1000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
1500	3.24	0.05	5.77	0.5	5.65	0.44	23.51	0.5	23.57	0.49
1500	3.29	0	5.6	0.55	5.61	0.49	23.68	0.49	23.62	0.5
1500	3.29	0	5.66	0.49	5.66	0.5	23.67	0.49	23.67	0.49
1500	3.29	0	5.66	0.49	5.6	0.44	23.67	0.55	23.61	0.44
1500	3.3	0	5.6	0.5	5.6	0.49	23.68	0.5	23.67	0.49
1500	3.24	0	5.66	0.49	5.6	0.5	23.67	0.49	23.62	0.5
1500	3.24	0	5.66	0.49	5.61	0.49	23.62	0.55	23.62	0.44
1500	3.24	0	5.65	0.5	5.6	0.44	23.67	0.49	23.62	0.5
1500	3.24	0	5.71	0.49	5.61	0.49	23.68	0.5	23.61	0.49
1500	3.3	0	5.66	0.49	5.6	0.5	23.67	0.49	23.67	0.5
1500	3.24	0	5.66	0.49	5.61	0.49	23.62	0.49	23.68	0.5
1500	3.25	0	5.65	0.5	5.65	0.5	23.62	0.49	23.67	0.5
1500	3.24	0	5.66	0.55	5.6	0.44	23.61	0.55	23.67	0.5
1500	3.24	0	5.77	0.5	5.65	0.44	23.67	0.55	23.62	0.44
1500	3.24	0	5.71	0.5	5.65	0.5	23.67	0.49	23.68	0.5
1500	3.24	0	5.66	0.55	5.6	0.49	23.67	0.55	23.67	0.5
1500	3.24	0.05	5.66	0.49	5.66	0.49	23.67	0.5	23.62	0.49
1500	3.24	0	5.65	0.55	5.61	0.49	23.68	0.49	23.67	0.5
1500	3.24	0.06	5.65	0.55	5.66	0.49	23.73	0.5	23.68	0.49
1500	3.24	0	5.65	0.5	5.66	0.49	23.67	0.49	23.67	0.5
1500	3.24	0	5.65	0.5	5.66	0.44	23.62	0.49	23.68	0.49
1500	3.24	0	5.72	0.49	5.66	0.44	0	0.49	0	0.5
1500	3.24	0	5.71	0.49	5.66	0.44	0	0.55	0	0.44
1500	3.24	0	5.66	0.49	5.61	0.44	0	0.54	0	0.44
1500	3.24	0	5.66	0.5	5.65	0.44	0	0.55	0	0.44
1500	3.24	0	5.76	0.5	5.66	0.49	0	0.5	0	0.49
1500	3.24	0	5.72	0.49	5.71	0.44	0	0.5	0	0.49
1500	3.24	0	5.71	0.5	5.66	0.44	0	0.49	0	0.49
1500	3.24	0	5.71	0.55	5.66	0.49	0	0.49	0	0.5
1500	3.24	0	5.65	0.5	5.66	0.43	0	0.55	0	0.44
1500	3.24	0	5.66	0.49	5.66	0.44	0	0.55	0	0.44
1500	3.24	0.05	5.66	0.55	5.65	0.5	0	0.49	0	0.5
1500	3.24	0	5.66	0.49	5.61	0.49	0	0.49	0	0.44
1500	3.24	0	5.66	0.49	5.61	0.43	0	0.55	0	0.44
1500	3.24	0	5.65	0.5	5.66	0.55	0	0.54	0	0.5
1500	3.24	0	5.66	0.49	5.66	0.44	0	0.49	0	0.5
1500	3.24	0	5.65	0.5	5.6	0.49	0	0.5	0	0.49
1500	3.24	0	5.72	0.49	5.66	0.44	0	0.55	0	0.49

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	Select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
1500	3.24	0	5.66	0.55	5.65	0.44	0	0.5	0	0.49
1500	3.24	0	5.66	0.49	5.66	0.49	0	0.5	0	0.49
1500	3.29	0	5.66	0.49	5.61	0.49	0	0.5	0	0.44
1500	3.24	0	5.66	0.49	5.66	0.5	0	0.49	0	0.49
1500	3.24	0	5.66	0.49	5.66	0.5	0	0.49	0	0.49
1500	3.24	0	5.71	0.5	5.66	0.49	0	0.49	0	0.5
1500	3.25	0.05	5.65	0.5	5.65	0.5	0	0.55	0	0.44
1500	3.25	0	5.65	0.5	5.65	0.5	0	0.49	0	0.5
1500	3.24	0.06	5.66	0.49	5.66	0.49	0	0.5	0	0.49
1500	3.18	0.06	5.71	0.5	5.65	0.5	0	0.55	0	0.49
1500	3.24	0	5.71	0.49	5.72	0.49	0	0.49	0	0.5
1500	3.24	0	5.66	0.49	5.6	0.44	0	0.5	0	0.49
Avg.	3.2458	0.0076	5.6734	0.5032	5.6404	0.474	23.65476	0.5102	23.64714	0.4818

Table 11: Time Evaluations When 1500 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
2000	4.34	0	8.01	0.66	7.97	0.6	31.8	0.66	31.74	0.66
2000	4.34	0	7.91	0.66	7.91	0.65	31.97	0.66	31.97	0.66
2000	4.34	0	7.97	0.66	7.91	0.6	31.91	0.66	31.91	0.66
2000	4.34	0.05	7.91	0.66	7.85	0.66	31.97	0.66	31.91	0.61
2000	4.33	0.06	7.91	0.66	7.91	0.61	31.97	0.66	31.91	0.65
2000	4.34	0	7.91	0.66	7.91	0.6	31.91	0.66	31.85	0.66
2000	4.34	0	7.91	0.66	7.86	0.6	31.91	0.66	31.91	0.66
2000	4.29	0	7.91	0.66	7.91	0.6	31.91	0.66	31.91	0.66
2000	4.34	0	7.96	0.72	7.85	0.66	31.92	0.66	31.91	0.66
2000	4.34	0	7.96	0.66	7.91	0.61	32.02	0.66	31.91	0.66
2000	4.34	0	7.91	0.66	7.86	0.65	32.02	0.66	31.91	0.66
2000	4.34	0.05	7.91	0.66	7.96	0.61	31.97	0.71	31.97	0.61
2000	4.34	0	7.96	0.66	7.91	0.6	31.91	0.66	31.91	0.66
2000	4.33	0	7.91	0.66	7.86	0.66	31.97	0.66	31.97	0.65
2000	4.28	0.05	7.91	0.66	7.91	0.66	32.02	0.66	31.91	0.6
2000	4.33	0	7.91	0.66	7.86	0.66	31.96	0.66	31.91	0.65
2000	4.28	0.06	8.02	0.66	7.91	0.66	31.96	0.66	31.91	0.6
2000	4.34	0	7.91	0.66	7.91	0.6	31.97	0.66	31.91	0.66
2000	4.28	0.06	7.97	0.66	7.85	0.66	31.96	0.66	31.91	0.66
2000	4.34	0	7.91	0.71	7.86	0.65	31.97	0.66	31.97	0.66
2000	4.34	0	7.91	0.66	7.96	0.6	31.97	0.66	31.96	0.66
2000	4.34	0	7.96	0.72	7.85	0.61	0	0.71	0	0.66
2000	4.34	0	7.97	0.66	7.91	0.66	0	0.65	0	0.66
2000	4.34	0	7.91	0.71	7.91	0.6	0	0.72	0	0.6
2000	4.34	0	7.9	0.66	7.86	0.71	0	0.66	0	0.66
2000	4.34	0	7.97	0.71	7.91	0.66	0	0.66	0	0.66
2000	4.34	0	7.9	0.72	7.91	0.66	0	0.66	0	0.65
2000	4.34	0	7.91	0.66	7.91	0.6	0	0.66	0	0.66
2000	4.29	0.05	7.96	0.72	7.91	0.6	0	0.71	0	0.66
2000	4.34	0	7.91	0.66	7.91	0.6	0	0.66	0	0.66
2000	4.34	0	7.97	0.65	7.97	0.6	0	0.66	0	0.66
2000	4.34	0	7.91	0.71	7.85	0.61	0	0.71	0	0.61
2000	4.34	0	7.91	0.66	7.96	0.6	0	0.66	0	0.66
2000	4.34	0	7.96	0.66	7.91	0.66	0	0.66	0	0.6
2000	4.34	0	7.96	0.66	7.91	0.66	0	0.66	0	0.66
2000	4.29	0.05	7.96	0.66	7.91	0.66	0	0.66	0	0.66
2000	4.28	0.06	7.91	0.66	7.91	0.6	0	0.72	0	0.6
2000	4.34	0	7.96	0.66	7.91	0.66	0	0.66	0	0.66

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
2000	4.34	0	7.96	0.66	7.91	0.66	0	0.66	0	0.66
2000	4.34	0	8.02	0.66	7.91	0.66	0	0.66	0	0.6
2000	4.34	0	7.85	0.66	7.86	0.66	0	0.71	0	0.66
2000	4.34	0	7.97	0.66	7.91	0.66	0	0.65	0	0.66
2000	4.28	0.06	7.97	0.66	7.96	0.61	0	0.65	0	0.66
2000	4.34	0	7.91	0.66	7.9	0.61	0	0.66	0	0.66
2000	4.34	0	7.91	0.66	7.91	0.66	0	0.6	0.06	0.66
2000	4.34	0	7.91	0.65	7.91	0.66	0	0.66	0	0.61
2000	4.28	0.06	7.97	0.65	7.91	0.61	0	0.66	0	0.66
2000	4.34	0	7.97	0.71	7.91	0.66	0	0.66	0	0.66
2000	4.34	0	7.96	0.66	7.91	0.66	0	0.66	0	0.6
2000	4.28	0	7.96	0.66	7.91	0.61	0	0.66	0	0.65
Avg.	4.328	0.0122	7.9376	0.6692	7.9032	0.6334	31.95095	0.6656	31.91571	0.6466

Table 12: Time Evaluations When 2000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
2500	5.43	0	10.66	0.88	10.32	0.77	40.31	0.83	40.32	0.82
2500	5.39	0.05	10.32	0.82	10.33	0.77	40.48	0.82	40.31	0.83
2500	5.44	0.05	10.33	0.82	10.33	0.77	40.43	0.87	40.37	0.77
2500	5.49	0	10.39	0.82	10.33	0.76	40.48	0.83	40.37	0.82
2500	5.44	0	10.32	0.83	10.27	0.82	40.48	0.83	40.32	0.82
2500	5.38	0	10.38	0.82	10.27	0.83	40.43	0.82	40.42	0.83
2500	5.39	0	10.43	0.83	10.38	0.77	40.42	0.82	40.37	0.82
2500	5.44	0	10.32	0.83	10.38	0.77	40.54	0.87	40.42	0.83
2500	5.39	0	10.49	0.82	10.44	0.82	40.48	0.83	40.43	0.82
2500	5.38	0.05	10.49	0.83	10.38	0.82	40.48	0.83	40.42	0.77
2500	5.44	0	10.38	0.82	10.39	0.76	40.48	0.83	40.37	0.82
2500	5.38	0	10.44	0.82	10.44	0.77	40.42	0.82	40.32	0.83
2500	5.38	0	10.49	0.82	10.49	0.77	40.42	0.83	40.42	0.82
2500	5.44	0	10.44	0.82	10.38	0.77	40.48	0.83	40.42	0.82
2500	5.44	0	10.43	0.83	10.38	0.82	40.48	0.83	40.37	0.82
2500	5.44	0	10.44	0.82	10.38	0.77	40.48	0.83	40.43	0.82
2500	5.44	0	10.43	0.82	10.39	0.76	40.48	0.83	40.37	0.82
2500	5.39	0.05	10.43	0.82	10.39	0.76	40.42	0.88	40.42	0.77
2500	5.38	0	10.39	0.82	10.38	0.77	40.42	0.82	40.37	0.83
2500	5.44	0	10.43	0.83	10.38	0.82	40.53	0.82	40.43	0.77
2500	5.38	0	10.38	0.99	10.33	0.77	40.48	0.87	40.43	0.83
2500	5.44	0	10.43	0.88	10.38	0.77	0	0.83	0	0.82
2500	5.44	0	10.44	0.82	10.44	0.82	0	0.82	0	0.77
2500	5.38	0	10.38	0.88	10.33	0.82	0	0.82	0	0.83
2500	5.38	0	10.43	0.83	10.43	0.77	0	0.83	0	0.82
2500	5.39	0	10.43	0.83	10.38	0.77	0	0.82	0	0.82
2500	5.44	0	10.44	0.82	10.44	0.82	0	0.83	0	0.76
2500	5.39	0.05	10.38	0.82	10.38	0.77	0	0.88	0	0.77
2500	5.39	0	10.49	0.82	10.38	0.82	0	0.83	0	0.77
2500	5.44	0	10.38	0.82	10.38	0.77	0	0.82	0	0.83
2500	5.38	0.06	10.43	0.83	10.38	0.82	0	0.83	0	0.82
2500	5.44	0	10.44	0.82	10.38	0.82	0	0.83	0	0.82
2500	5.38	0.06	10.49	0.82	10.5	0.76	0	0.83	0	0.82
2500	5.38	0	10.44	0.87	10.39	0.82	0	0.82	0	0.83
2500	5.39	0	10.49	0.82	10.38	0.82	0	0.83	0	0.77
2500	5.38	0	10.38	0.88	10.33	0.82	0	0.83	0	0.82
2500	5.38	0	10.44	0.82	10.33	0.76	0	0.88	0	0.77
2500	5.39	0	10.49	0.82	10.44	0.82	0	0.83	0	0.82

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
2500	5.38	0.05	10.44	0.88	10.38	0.77	0	0.82	0	0.83
2500	5.38	0	10.43	0.88	10.38	0.77	0	0.83	0	0.82
2500	5.44	0	10.38	0.82	10.38	0.83	0	0.82	0	0.82
2500	5.38	0.06	10.49	0.83	10.43	0.77	0	0.82	0	0.83
2500	5.38	0.06	10.38	0.83	10.32	0.77	0	0.83	0	0.82
2500	5.44	0	10.38	0.93	10.39	0.82	0	0.82	0	0.83
2500	5.43	0	10.44	0.88	10.38	0.82	0	0.83	0	0.82
2500	5.44	0	10.38	0.82	10.38	0.77	0	0.83	0	0.82
2500	5.38	0.05	10.44	0.82	10.38	0.77	0	0.88	0	0.77
2500	5.38	0	10.44	0.88	10.38	0.77	0	0.82	0	0.83
2500	5.44	0	10.43	0.88	10.44	0.76	0	0.83	0	0.82
2500	5.38	0.06	10.38	0.88	10.33	0.82	0	0.83	0	0.82
Avg.	5.4076	0.013	10.4244	0.8408	10.38	0.789	40.4581	0.8332	40.38571	0.8114

Table 13: Time Evaluations When 2500 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
3000	6.48	0	12.96	0.99	12.69	0.93	48.61	1.04	48.61	0.94
3000	6.48	0.05	12.69	1.05	12.63	0.93	48.83	0.99	48.82	0.99
3000	6.48	0.05	12.75	0.99	12.74	0.93	48.83	0.99	48.72	0.99
3000	6.48	0.06	12.74	1.04	12.58	0.99	48.77	1.04	48.78	0.94
3000	6.49	0.05	12.74	0.99	12.63	0.93	48.83	1.05	48.72	0.93
3000	6.43	0.05	12.74	0.99	12.69	0.93	48.77	1.04	48.61	0.94
3000	6.48	0.06	12.69	0.99	12.68	0.94	48.77	1.04	48.78	0.99
3000	6.48	0	12.74	0.94	12.68	0.94	48.78	0.99	48.61	0.93
3000	6.43	0.05	12.74	0.99	12.69	0.93	48.83	0.99	48.77	0.94
3000	6.48	0	12.91	0.99	12.85	0.93	48.88	0.99	48.72	0.94
3000	6.48	0.06	12.74	1.04	12.69	0.94	48.82	1.04	48.67	0.93
3000	6.48	0.05	12.74	0.99	12.75	0.93	48.77	1.04	48.77	0.94
3000	6.49	0.05	12.79	0.94	12.74	0.93	48.83	0.99	48.72	0.99
3000	6.49	0.05	12.79	1.05	12.69	0.98	48.83	0.99	48.71	0.99
3000	6.49	0.05	12.74	0.99	12.79	0.94	48.88	0.99	48.77	0.93
3000	6.49	0.05	12.74	0.99	12.69	0.93	48.82	1.04	48.78	0.94
3000	6.42	0.06	12.75	1.04	12.69	0.99	48.89	0.99	48.78	0.93
3000	6.48	0	12.74	0.99	12.69	0.93	48.77	1.05	48.77	0.93
3000	6.48	0.05	12.69	0.99	12.63	0.94	48.77	1.04	48.77	0.93
3000	6.48	0	12.85	0.99	12.8	0.93	48.89	1.04	48.82	0.99
3000	6.48	0	12.8	0.93	12.74	0.94	48.83	0.99	48.78	0.98
3000	6.49	0	12.74	0.99	12.68	0.94	0	0.99	0	0.99
3000	6.48	0	12.8	0.99	12.79	0.94	0	0.99	0	0.93
3000	6.48	0.05	12.75	0.99	12.68	0.94	0	0.99	0	0.98
3000	6.48	0.06	12.68	0.99	12.64	0.93	0	1.04	0	0.94
3000	6.42	0.06	12.75	0.99	12.63	0.93	0	1.05	0	0.93
3000	6.48	0	12.85	1.05	12.79	0.94	0	1.1	0	0.93
3000	6.48	0	12.86	0.99	12.74	0.99	0	0.99	0	0.93
3000	6.48	0	12.75	0.99	12.68	0.94	0	0.99	0	0.98
3000	6.43	0.05	12.85	1.05	12.79	0.94	0	1.1	0	0.93
3000	6.48	0	12.79	0.99	12.75	0.93	0	0.99	0	0.99
3000	6.54	0	12.8	0.98	12.8	0.94	0	0.98	0	0.99
3000	6.54	0	12.63	0.99	12.74	0.94	0	0.98	0	0.99
3000	6.53	0	12.75	0.98	12.8	0.94	0	0.98	0	0.94
3000	6.48	0	12.86	0.99	12.74	0.99	0	0.99	0	0.93
3000	6.48	0	12.8	0.99	12.74	1.05	0	1.04	0	0.93
3000	6.48	0	12.74	0.93	12.69	0.94	0	0.99	0	0.98
3000	6.54	0	12.74	0.99	12.74	0.93	0	1.05	0	0.93

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
3000	6.48	0.05	12.8	0.99	12.74	0.93	0	0.99	0	0.94
3000	6.48	0.06	12.69	1.04	12.63	0.99	0	1.04	0	0.99
3000	6.49	0	12.74	0.99	12.68	0.94	0	0.99	0	0.99
3000	6.48	0.05	12.86	0.99	12.74	0.99	0	0.99	0	0.93
3000	6.49	0.05	12.68	1.05	12.68	0.94	0	1.04	0	0.99
3000	6.53	0	12.75	0.98	12.75	0.93	0	0.99	0	0.93
3000	6.48	0	12.8	0.99	12.74	0.93	0	0.99	0	0.99
3000	6.48	0	12.8	0.99	12.74	0.94	0	0.98	0	0.99
3000	6.48	0	12.85	0.99	12.8	0.93	0	1.04	0	0.94
3000	6.48	0.05	12.86	1.04	12.8	0.93	0	0.99	0	0.99
3000	6.42	0.06	12.8	1.04	12.69	0.99	0	1.04	0	0.99
3000	6.54	0	12.79	0.99	12.74	0.94	0	0.99	0	0.93
Avg.	6.4818	0.0266	12.7726	0.997	12.7182	0.9458	48.80952	1.0134	48.73714	0.9566

Table 14: Time Evaluations When 3000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
4000	8.67	0.06	18.84	1.27	18.23	1.27	66.02	1.31	66.13	1.27
4000	8.63	0.05	18.01	1.32	17.96	1.32	66.29	1.37	66.24	1.27
4000	8.68	0	18.07	1.32	17.96	1.32	66.19	1.37	66.08	1.27
4000	8.62	0.06	18.18	1.32	18.07	1.26	66.29	1.32	66.3	1.27
4000	8.67	0	18.13	1.26	18.07	1.21	66.19	1.38	66.19	1.26
4000	8.63	0.05	18.12	1.37	18.07	1.32	66.24	1.32	66.18	1.32
4000	8.63	0.05	18.07	1.37	18.02	1.26	66.24	1.32	66.35	1.26
4000	8.63	0	18.01	1.32	17.91	1.26	66.35	1.32	66.35	1.32
4000	8.68	0	18.18	1.32	18.12	1.32	66.3	1.37	66.24	1.27
4000	8.68	0	18.18	1.37	18.07	1.32	66.3	1.37	66.19	1.32
4000	8.62	0	18.18	1.32	18.02	1.31	66.29	1.32	66.07	1.27
4000	8.63	0	18.18	1.31	18.13	1.26	66.19	1.32	66.07	1.26
4000	8.62	0.06	18.34	1.32	18.29	1.27	66.3	1.42	66.13	1.32
4000	8.62	0.06	18.34	1.32	18.35	1.26	66.3	1.32	66.19	1.26
4000	8.67	0	18.18	1.27	18.12	1.27	66.35	1.31	66.19	1.27
4000	8.62	0.05	18.24	1.43	18.23	1.27	66.24	1.37	66.07	1.32
4000	8.68	0	18.29	1.32	18.18	1.32	66.24	1.32	66.24	1.32
4000	8.63	0.05	18.29	1.32	18.12	1.26	66.24	1.32	66.24	1.27
4000	8.62	0.06	18.18	1.38	18.01	1.32	66.24	1.37	66.29	1.27
4000	8.62	0.06	18.29	1.32	18.18	1.26	66.35	1.32	66.18	1.27
4000	8.62	0.06	18.29	1.32	18.18	1.26	66.29	1.32	66.24	1.27
4000	8.67	0	18.24	1.32	18.12	1.38	0	1.31	0	1.27
4000	8.68	0	18.18	1.32	18.18	1.27	0	1.31	0	1.27
4000	8.62	0.06	18.18	1.32	18.07	1.26	0	1.32	0	1.27
4000	8.68	0	18.07	1.32	18.01	1.32	0	1.38	0	1.31
4000	8.63	0.05	18.23	1.38	18.18	1.26	0	1.43	0	1.26
4000	8.62	0	18.23	1.38	18.12	1.27	0	1.31	0	1.27
4000	8.68	0	18.02	1.37	17.91	1.32	0	1.31	0	1.27
4000	8.62	0.06	18.29	1.38	18.18	1.26	0	1.37	0	1.38
4000	8.68	0	18.24	1.32	18.18	1.26	0	1.32	0	1.26
4000	8.63	0.05	18.07	1.31	18.02	1.21	0	1.32	0	1.31
4000	8.62	0.06	18.24	1.31	18.24	1.26	0	1.38	0	1.31
4000	8.62	0.06	18.13	1.37	18.07	1.26	0	1.32	0	1.27
4000	8.68	0	18.24	1.31	18.19	1.26	0	1.32	0	1.26
4000	8.63	0	18.23	1.32	18.13	1.31	0	1.38	0	1.32
4000	8.62	0	18.13	1.31	18.07	1.27	0	1.32	0	1.26
4000	8.62	0.06	18.13	1.32	18.07	1.26	0	1.32	0	1.26
4000	8.62	0.06	18.18	1.32	18.07	1.26	0	1.43	0	1.32

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
4000	8.68	0	18.13	1.37	18.12	1.27	0	1.37	0	1.26
4000	8.62	0.06	18.13	1.31	18.02	1.26	0	1.32	0	1.32
4000	8.62	0	18.18	1.32	18.13	1.26	0	1.37	0	1.32
4000	8.63	0	18.29	1.37	18.29	1.26	0	1.38	0	1.26
4000	8.62	0	18.23	1.32	18.18	1.21	0	1.37	0	1.27
4000	8.68	0	18.23	1.32	18.18	1.26	0	1.32	0	1.27
4000	8.62	0.06	18.18	1.37	18.13	1.32	0	1.37	0	1.26
4000	8.62	0.06	18.13	1.37	18.02	1.26	0	1.32	0	1.26
4000	8.63	0	18.18	1.31	18.13	1.26	0	1.32	0	1.26
4000	8.63	0.05	18.29	1.26	18.23	1.32	0	1.38	0	1.26
4000	8.62	0.06	18.13	1.32	18.07	1.26	0	1.32	0	1.26
4000	8.67	0	18.29	1.32	18.18	1.32	0	1.32	0	1.26
Avg.	8.6406	0.0284	18.2002	1.3302	18.1156	1.2774	66.25905	1.3434	66.1981	1.2812

Table 15: Time Evaluations When 4000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
5000	10.82	0.05	25.54	1.65	24.55	1.6	84.65	1.64	84.64	1.6
5000	10.82	0.06	24.77	1.65	24.71	1.54	84.97	1.65	88.16	1.59
5000	10.82	0.06	24.66	1.65	24.6	1.54	84.7	1.7	84.75	1.6
5000	10.77	0.05	24.83	1.64	24.72	1.59	84.75	1.71	84.86	1.59
5000	10.76	0.06	24.61	1.7	24.5	1.59	84.64	1.65	84.64	1.59
5000	10.77	0.05	24.88	1.59	24.77	1.6	84.91	1.59	84.8	1.59
5000	10.76	0.06	24.72	1.65	24.6	1.65	84.81	1.7	84.81	1.6
5000	10.77	0.05	24.49	1.82	24.49	1.6	84.86	1.64	84.86	1.65
5000	10.82	0	24.77	1.65	24.72	1.59	85.03	1.7	84.64	1.6
5000	10.76	0.06	24.99	1.65	24.94	1.59	84.86	1.59	84.86	1.6
5000	10.82	0	24.72	1.65	24.6	1.65	85.03	1.65	84.86	1.59
5000	10.82	0.06	25.05	1.7	25.04	1.6	84.8	1.7	84.7	1.59
5000	10.76	0	24.83	1.65	24.71	1.6	84.86	1.64	84.8	1.54
5000	10.76	0.06	24.94	1.65	24.77	1.65	84.86	1.7	84.75	1.59
5000	10.82	0.05	25.21	1.76	25.1	1.6	84.75	1.64	84.58	1.6
5000	10.82	0.05	24.83	1.65	24.82	1.54	84.92	1.65	84.75	1.59
5000	10.82	0.05	24.99	1.65	24.99	1.7	84.86	1.71	84.64	1.53
5000	10.76	0.06	24.88	1.7	24.83	1.54	84.92	1.7	84.69	1.59
5000	10.82	0.05	24.77	1.65	24.72	1.53	84.81	1.65	85.02	1.59
5000	10.82	0	24.99	1.65	24.88	1.65	84.92	1.76	84.91	1.53
5000	10.82	0	24.88	1.71	24.66	1.59	84.97	1.65	84.86	1.59
5000	10.82	0	24.83	1.65	24.66	1.59	0	1.65	0	1.59
5000	10.82	0.05	25.05	1.59	25.05	1.53	0	1.65	0	1.6
5000	10.82	0.05	24.94	1.7	24.77	1.59	0	1.65	0	1.6
5000	10.82	0.06	24.72	1.64	24.66	1.54	0	1.65	0	1.59
5000	10.82	0.06	24.82	1.65	24.77	1.54	0	1.65	0	1.59
5000	10.82	0	24.99	1.65	24.83	1.59	0	1.7	0	1.6
5000	10.82	0.05	24.77	1.71	24.66	1.59	0	1.65	0	1.65
5000	10.77	0.05	25.04	1.65	24.99	1.54	0	1.65	0	1.59
5000	10.82	0.06	24.93	1.65	24.88	1.59	0	1.71	0	1.59
5000	10.82	0.06	24.83	1.7	24.77	1.54	0	1.65	0	1.59
5000	10.76	0.06	24.94	1.65	24.94	1.53	0	1.71	0	1.53
5000	10.77	0.05	24.99	1.64	24.89	1.64	0	1.71	0	1.64
5000	10.83	0.05	24.99	1.7	24.77	1.59	0	1.65	0	1.59
5000	10.77	0.05	24.93	1.7	24.83	1.59	0	1.65	0	1.59
5000	10.82	0.05	24.67	1.64	24.67	1.64	0	1.65	0	1.59
5000	10.76	0.06	24.77	1.71	24.71	1.59	0	1.65	0	1.6
5000	10.82	0.05	24.77	1.65	24.72	1.54	0	1.7	0	1.54

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
5000	10.82	0	24.77	1.65	24.71	1.54	0	1.65	0	1.59
5000	10.77	0.05	24.72	1.7	24.83	1.59	0	1.65	0	1.59
5000	10.82	0.06	24.99	1.81	24.88	1.54	0	1.7	0	1.54
5000	10.82	0	24.77	1.65	24.77	1.53	0	1.65	0	1.6
5000	10.82	0.05	24.77	1.65	24.72	1.65	0	1.7	0	1.59
5000	10.82	0	24.77	1.76	24.66	1.65	0	1.65	0	1.59
5000	10.82	0.06	24.83	1.64	24.77	1.6	0	1.59	0	1.59
5000	10.83	0	24.77	1.64	24.72	1.65	0	1.65	0	1.59
5000	10.82	0.06	24.83	1.7	24.77	1.59	0	1.71	0	1.59
5000	10.82	0	25	1.7	24.93	1.6	0	1.64	0	1.6
5000	10.82	0	24.93	1.65	24.88	1.65	0	1.7	0.06	1.53
5000	10.82	0	24.93	1.71	24.77	1.59	0	1.65	0	1.59
Avg.	10.8038	0.0394	24.8682	1.6722	24.774	1.5896	84.85143	1.6658	84.93524	1.588

Table 16: Time Evaluations When 5000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
10000	22.85	0.11	58.71	3.3	53	3.19	173.56	3.29	173.4	3.19
10000	21.64	0.05	53.44	3.35	53.34	3.13	173.35	3.29	173.18	3.19
10000	21.64	0.06	53.6	3.35	53.45	3.18	173.23	3.35	173.29	3.19
10000	21.58	0.06	53.89	3.24	53.71	3.25	173.35	3.35	173.46	3.18
10000	21.58	0.11	53.78	3.29	53.44	3.19	173.18	3.3	173.62	3.24
10000	21.53	0.05	53.61	3.35	53.39	3.13	173.57	3.35	172.91	3.13
10000	21.53	0.11	53.44	3.24	53.22	3.19	173.45	3.35	173.13	3.18
10000	21.64	0.06	53.17	3.29	53.17	3.24	173.46	3.3	172.96	3.18
10000	21.58	0.06	54.1	3.3	54.21	3.13	173.23	3.3	173.29	3.13
10000	21.58	0.06	53.72	3.4	53.61	3.24	173.79	3.3	173.57	3.18
10000	21.59	0.05	53.71	3.24	53.56	3.18	173.67	3.35	173.4	3.19
10000	21.64	0.06	54.21	3.35	54.05	3.13	173.23	3.29	172.85	3.19
10000	21.59	0.05	53.66	3.35	53.55	3.13	173.29	3.3	172.9	3.24
10000	21.59	0.05	54.1	3.29	53.72	3.19	173.62	3.29	173.57	3.13
10000	21.64	0.06	53.82	3.35	53.94	3.19	173.29	3.35	172.9	3.18
10000	21.64	0.06	53.71	3.36	53.6	3.13	173.29	3.3	173.24	3.18
10000	21.64	0.05	53.39	3.3	53.44	3.24	173.4	3.3	172.85	3.18
10000	21.64	0.06	54.1	3.3	53.82	3.19	173.51	3.3	173.62	3.13
10000	21.7	0.06	53.82	3.35	53.72	3.24	173.13	3.24	173.13	3.19
10000	21.59	0.05	53.88	3.3	53.71	3.3	173.67	3.3	173.18	3.24
10000	21.59	0.05	53.6	3.35	53.45	3.13	173.78	3.29	173.45	3.19
10000	21.59	0.05	53.49	3.36	53.33	3.13	0	3.35	0	3.18
10000	21.53	0.06	53.82	3.35	53.72	3.24	0	3.3	0	3.18
10000	21.59	0.05	53.6	3.24	53.67	3.13	0	3.29	0	3.13
10000	21.58	0.06	53.28	3.35	53.17	3.13	0	3.35	0	3.19
10000	21.53	0.05	54.21	3.41	54.04	3.3	0	3.24	0	3.19
10000	21.58	0.11	53.77	3.24	53.67	3.18	0	3.35	0	3.24
10000	21.64	0.05	53.66	3.35	54.05	3.24	0	3.35	0	3.19
10000	21.64	0.06	54.21	3.35	53.99	3.19	0	3.29	0	3.25
10000	21.59	0.05	53.83	3.24	53.66	3.13	0	3.35	0	3.24
10000	21.58	0.11	54.27	3.35	53.77	3.13	0	3.3	0	3.19
10000	21.59	0.05	53.72	3.35	53.6	3.24	0	3.3	0	3.13
10000	21.58	0.06	53.83	3.3	53.66	3.24	0	3.35	0	3.19
10000	21.59	0.05	53.99	3.3	53.88	3.13	0	3.35	0	3.13
10000	21.64	0.06	53.77	3.41	53.66	3.13	0	3.35	0	3.13
10000	21.59	0.11	53.61	3.24	53.49	3.19	0	3.3	0	3.18
10000	21.59	0.05	53.49	3.3	53.55	3.24	0	3.24	0	3.19
10000	21.58	0.06	53.77	3.35	53.5	3.13	0	3.3	0	3.13

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
10000	21.59	0.05	53.72	3.35	53.55	3.13	0	3.35	0	3.19
10000	21.59	0.05	53.72	3.29	53.77	3.24	0	3.25	0	3.24
10000	21.59	0.05	53.6	3.41	53.55	3.13	0	3.3	0	3.18
10000	21.64	0.05	53.61	3.35	53.44	3.13	0	3.3	0	3.24
10000	21.58	0.06	53.5	3.29	53.28	3.24	0	3.3	0	3.13
10000	21.58	0.06	53.89	3.35	53.66	3.19	0	3.29	0	3.19
10000	21.59	0.05	53.77	3.41	53.6	3.19	0	3.29	0	3.19
10000	21.64	0.05	53.45	3.29	53.17	3.19	0	3.35	0	3.18
10000	21.59	0.05	53.61	3.35	53.44	3.13	0	3.35	0	3.19
10000	21.58	0.06	53.89	3.4	53.66	3.19	0	3.24	0	3.18
10000	21.59	0.05	53.55	3.24	53.45	3.18	0	3.35	0	3.19
10000	21.59	0.05	53.66	3.35	53.55	3.19	0	3.24	0	3.24
Avg.	21.6232	0.061	53.835	3.3244	53.5926	3.183	173.431	3.3092	173.2333	3.1846

Table 17: Time Evaluations When 10,000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
15000	33.5	0.11	88.54	5	79.8	4.67	264.52	5	259.46	4.72
15000	32.41	0.11	80.08	5.05	79.59	4.83	260.02	4.89	259.52	4.73
15000	32.41	0.11	79.92	4.88	80.14	4.78	259.68	4.89	259.03	4.78
15000	32.4	0.11	80.47	4.94	80.25	4.78	260.13	4.88	259.74	4.73
15000	32.46	0.11	80.74	5	80.57	4.78	260.12	4.94	259.57	4.73
15000	32.46	0.05	81.29	5	80.69	4.72	260.41	4.89	259.96	4.72
15000	32.47	0.05	80.13	4.95	80.35	4.67	260.18	4.89	259.53	4.72
15000	32.46	0.11	80.63	5	80.63	4.89	260.12	4.89	259.8	4.72
15000	32.46	0.06	79.97	5	80.19	4.78	260.18	4.89	259.96	4.72
15000	32.41	0.11	80.52	4.94	80.19	4.67	260.84	4.95	259.63	4.72
15000	32.29	0.11	80.52	5	80.63	4.84	260.13	4.89	259.53	4.66
15000	32.4	0.11	80.53	4.94	80.63	4.78	260.24	4.94	259.74	4.72
15000	32.35	0.11	80.63	4.89	80.36	4.83	260.18	5	259.85	4.72
15000	32.4	0.11	80.63	5	80.52	4.73	260.57	4.94	260.18	4.78
15000	32.41	0.11	80.85	4.94	80.91	4.83	260.35	4.94	259.47	4.78
15000	32.47	0.05	81.28	4.89	80.58	4.72	260.29	4.89	259.63	4.72
15000	32.41	0.11	80.46	4.95	80.24	4.67	259.3	5	259.19	4.72
15000	32.35	0.06	80.41	5	80.13	4.89	260.51	4.89	260.02	4.72
15000	32.46	0.11	80.35	4.95	80.13	4.78	260.35	4.95	260.4	4.72
15000	32.4	0.06	80.69	4.89	80.3	4.72	260.18	4.89	259.74	4.83
15000	32.4	0.06	80.41	4.95	80.13	4.78	260.07	4.95	260.18	4.77
15000	32.36	0.11	80.35	5	79.92	4.72	0	5	0	4.72
15000	32.41	0.11	80.74	4.94	80.63	4.73	0	4.94	0	4.72
15000	32.41	0.05	80.46	4.95	80.3	4.83	0	4.89	0	4.72
15000	32.4	0.11	80.91	5	80.13	4.78	0	4.83	0	4.84
15000	32.46	0.06	80.35	5.06	80.35	4.67	0	4.94	0	4.73
15000	32.35	0.11	80.53	4.99	80.47	4.78	0	4.89	0	4.72
15000	32.35	0.11	80.63	5	80.41	4.78	0	4.94	0	4.72
15000	32.36	0.1	80.74	4.99	80.31	4.77	0	4.89	0	4.73
15000	32.46	0.05	80.69	5.05	80.36	4.67	0	4.99	0	4.73
15000	32.41	0.1	81.68	5.05	81.29	4.83	0	4.94	0	4.73
15000	32.4	0.11	80.03	4.94	80.25	4.78	0	4.94	0	4.72
15000	32.35	0.11	80.52	5.05	80.47	4.72	0	4.89	0	4.78
15000	32.4	0.11	80.69	4.94	80.41	4.78	0	4.94	0	4.78
15000	32.41	0.11	80.08	5	80.52	4.72	0	5	0	4.78
15000	32.41	0.05	80.63	5.05	80.08	4.73	0	4.88	0	4.67
15000	32.35	0.11	80.46	4.95	80.3	4.78	0	4.89	0	4.72
15000	32.4	0.06	80.86	4.88	80.47	4.78	0	4.89	0	4.72

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
15000	32.36	0.1	80.35	4.95	80.3	4.67	0	4.88	0	4.73
15000	32.46	0.06	80.3	5	80.19	4.83	0	5	0	4.72
15000	32.35	0.05	80.53	5.05	80.47	4.72	0	4.94	0	4.73
15000	32.4	0.06	80.63	5	80.25	4.83	0	4.89	0	4.72
15000	32.41	0.11	81.29	5	81.12	4.84	0	4.99	0	4.73
15000	32.46	0.11	80.74	4.94	80.52	4.78	0	4.89	0	4.72
15000	32.46	0.11	80.47	5.05	80.19	4.83	0	4.89	0	4.84
15000	32.41	0.05	81.23	5	80.47	4.67	0	4.88	0	4.84
15000	32.4	0.11	80.58	5.05	80.41	4.83	0	4.89	0	4.73
15000	32.36	0.11	81.67	5	81.4	4.83	0	4.89	0	4.72
15000	32.46	0.11	80.35	5	80.36	4.72	0	4.89	0	4.72
15000	32.35	0.06	80.58	4.94	80.36	4.77	0	5	0	4.78
Avg.	32.427	0.0908	80.7624	4.9796	80.3934	4.7662	260.3986	4.922	259.7205	4.7388

Table 18: Time Evaluations When 15,000 Data Pairs were Compared

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
20000	44.71	0.11	122.98	6.64	106.28	6.32	346.96	6.59	346.52	6.26
20000	43.28	0.17	106.5	6.64	106.18	6.42	346.63	6.54	346.69	6.31
20000	43.12	0.16	107.38	6.53	106.94	6.32	347.02	6.54	346.8	6.26
20000	43.17	0.17	106.61	6.7	106.23	6.37	346.75	6.53	346.52	6.21
20000	43.17	0.11	107.71	6.54	107.43	6.37	346.63	6.48	346.53	6.27
20000	43.17	0.11	106.94	6.64	106.01	6.32	347.18	6.53	345.76	6.37
20000	43.11	0.11	106.94	6.65	106.44	6.38	346.97	6.59	345.04	6.26
20000	43.23	0.11	106.28	6.53	106.07	6.31	346.8	6.48	346.15	6.32
20000	43.23	0.1	106.89	6.64	107.44	6.31	347.35	6.54	346.52	6.26
20000	43.23	0.11	106.66	6.59	106.39	6.38	346.79	6.48	345.92	6.31
20000	43.23	0.11	108.31	6.59	108.09	6.32	347.24	6.48	347.18	6.32
20000	43.23	0.11	107.38	6.65	106.83	6.31	347.19	6.59	346.8	6.27
20000	43.22	0.17	107.6	6.59	107.22	6.42	346.8	6.48	346.42	6.38
20000	43.23	0.11	107.1	6.65	106.83	6.37	345.98	6.59	345.81	6.32
20000	43.17	0.11	107.93	6.59	108.26	6.37	345.97	6.54	345.92	6.26
20000	43.23	0.11	107.98	6.65	107.7	6.32	347.07	6.59	346.8	6.26
20000	43.23	0.11	107.76	6.59	107.38	6.32	346.86	6.59	345.7	6.32
20000	43.11	0.17	107.6	6.59	107.49	6.37	346.8	6.6	347.02	6.26
20000	43.12	0.11	107.43	6.71	107.48	6.32	347.13	6.65	345.81	6.37
20000	43.17	0.11	107.05	6.76	106.5	6.37	346.42	6.54	345.54	6.31
20000	43.17	0.17	107.38	6.65	107.05	6.37	346.2	6.53	346.74	6.32
20000	43.12	0.11	106.72	6.59	106.39	6.37	0	6.54	0	6.26
20000	43.28	0.11	107.44	6.59	107.1	6.32	0	6.59	0	6.37
20000	43.12	0.17	107.38	6.59	106.72	6.37	0	6.59	0	6.26
20000	43.17	0.11	106.88	6.65	106.94	6.37	0	6.48	0	6.32
20000	43.11	0.11	106.56	6.7	106.34	6.37	0	6.53	0	6.27
20000	43.23	0.11	107.65	6.65	107.7	6.32	0	6.59	0	6.32
20000	43.12	0.11	107.65	6.59	107.82	6.32	0	6.48	0	6.31
20000	43.23	0.11	106.88	6.54	106.55	6.27	0	6.53	0	6.37
20000	43.28	0.11	107.82	6.59	107.6	6.31	0	6.54	0	6.26
20000	43.17	0.11	107.44	6.7	107.82	6.37	0	6.54	0	6.31
20000	43.22	0.17	107.71	6.59	107.44	6.26	0	6.54	0	6.37
20000	43.28	0.11	107.49	6.59	107.11	6.37	0	6.54	0	6.26
20000	43.12	0.11	106.83	6.59	106.44	6.32	0	6.59	0	6.32
20000	43.17	0.11	107.88	6.64	107.77	6.31	0	6.6	0	6.48
20000	43.28	0.11	107.65	6.65	107.65	6.27	0	6.64	0	6.32
20000	43.28	0.11	107.44	6.59	107.16	6.37	0	6.59	0	6.37
20000	43.23	0.11	107.16	6.59	106.89	6.26	0	6.53	0	6.27

Pentium III, Windows, 256 meg RAM										
	storage time only	Local repository manipulation and evaluation								
		no connect	single connect				multiple connects			
number items in data set	load data and results into db	data in local memory	select each and do comparison	select all and compare	select each result from db	select all results from db	select each and do comparison	select all and compare	select each result from db	select all results from db
20000	43.06	0.16	107.66	6.64	107.44	6.31	0	6.6	0	6.26
20000	43.17	0.11	106.94	6.59	106.62	6.31	0	6.54	0	6.26
20000	43.17	0.16	106.94	6.65	106.39	6.32	0	6.53	0	6.32
20000	43.18	0.11	107.37	6.65	107.11	6.42	0	6.48	0	6.32
20000	43.11	0.11	107.82	6.59	107.71	6.27	0	6.59	0	6.26
20000	43.17	0.17	107.49	6.64	107.66	6.37	0	6.54	0	6.26
20000	43.28	0.11	107.38	6.59	107.1	6.26	0	6.6	0	6.31
20000	43.28	0.11	108.1	6.64	107.66	6.31	0	6.54	0	6.32
20000	43.17	0.11	107.32	6.59	107.11	6.31	0	6.65	0	6.32
20000	43.17	0.11	107.16	6.64	107.38	6.32	0	6.48	0	6.43
20000	43.12	0.11	106.93	6.65	106.78	6.37	0	6.59	0	6.31
20000	43.22	0.11	107.27	6.65	106.77	6.37	0	6.54	0	6.32
Avg.	43.2208	0.1224	107.6274	6.6214	107.0682	6.337	346.7971	6.552	346.2948	6.307

Table 19: Time Evaluations When 20000 Data Pairs were Compared

APPENDIX H

Additional Graphs from Pre-Analysis Results

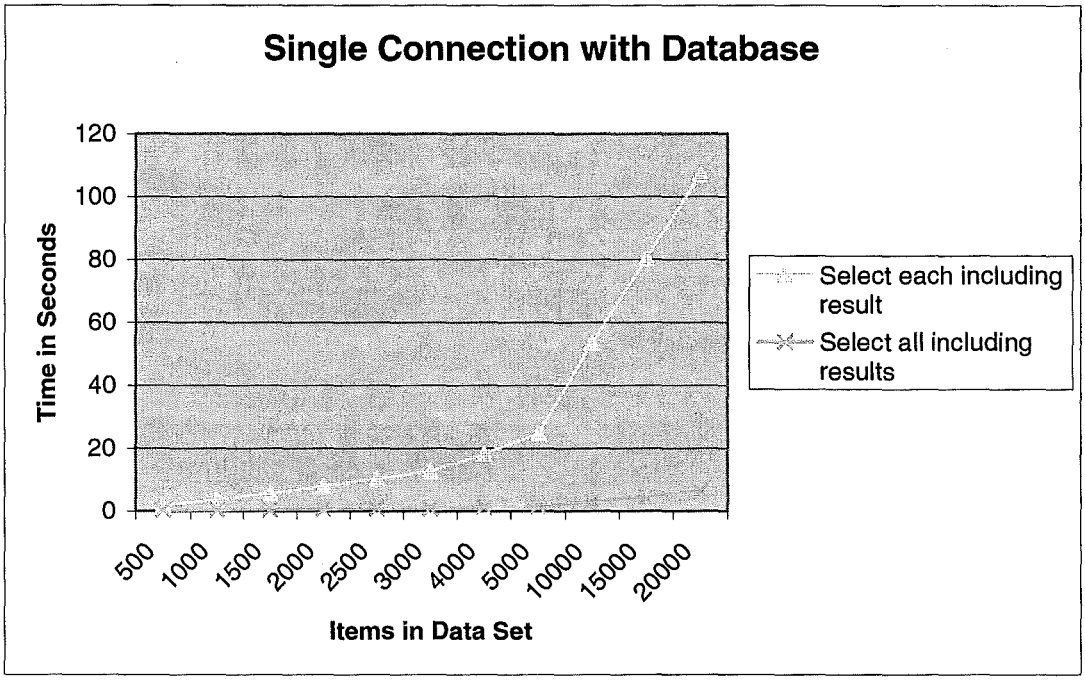


Figure 16: Comparison of Selecting All Data Sets with Previously Evaluated Results using Multiple Selects Versus a Single Select Statement During a Single Connection to the Database

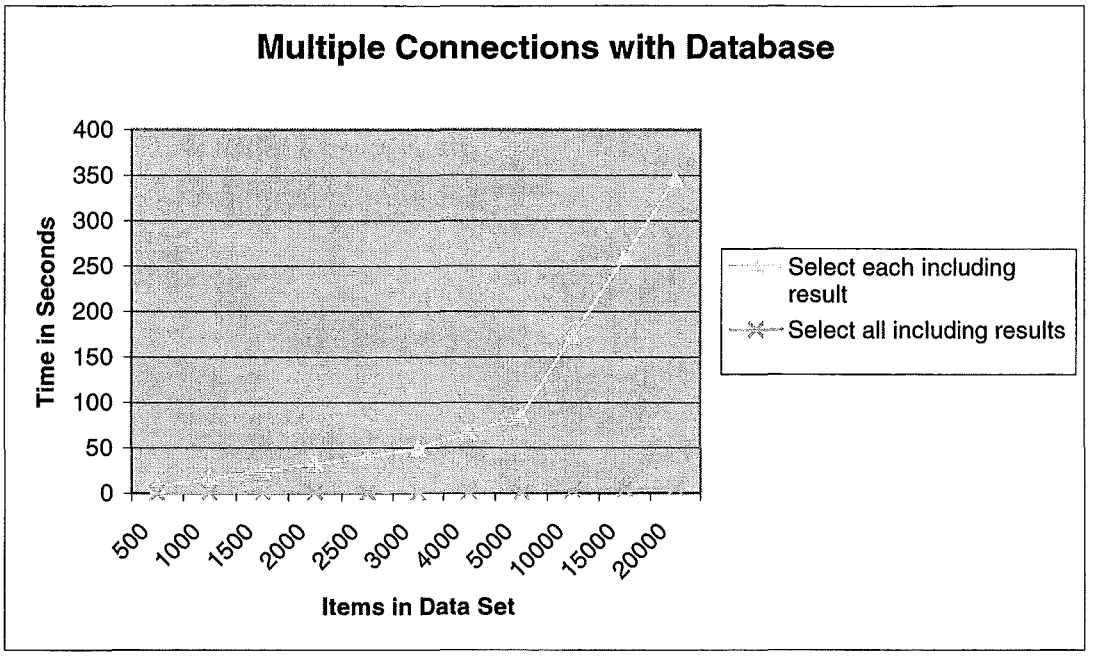


Figure 17: Comparison of Selecting All Data Sets with Previously Evaluated Results using Multiple Selects Versus a Single Select Statement During Multiple Connections to the Database

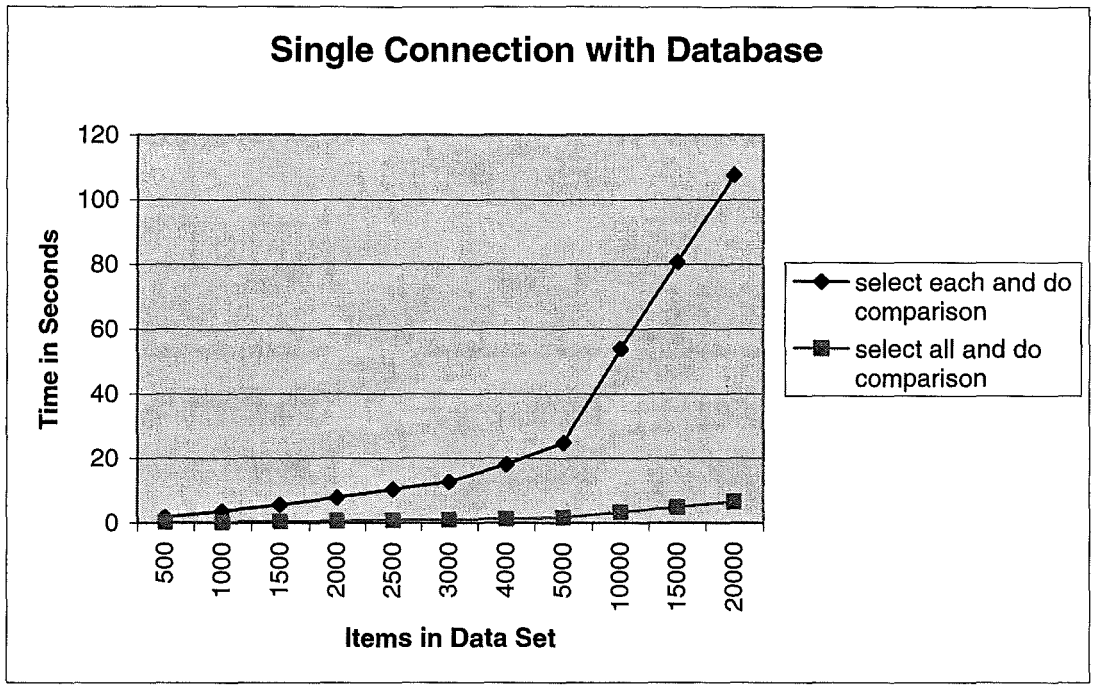


Figure 18: Comparison of Selecting All Data Sets and Performing Evaluations using Multiple Selects Versus a Single Select Statement During a Single Connection to the Database

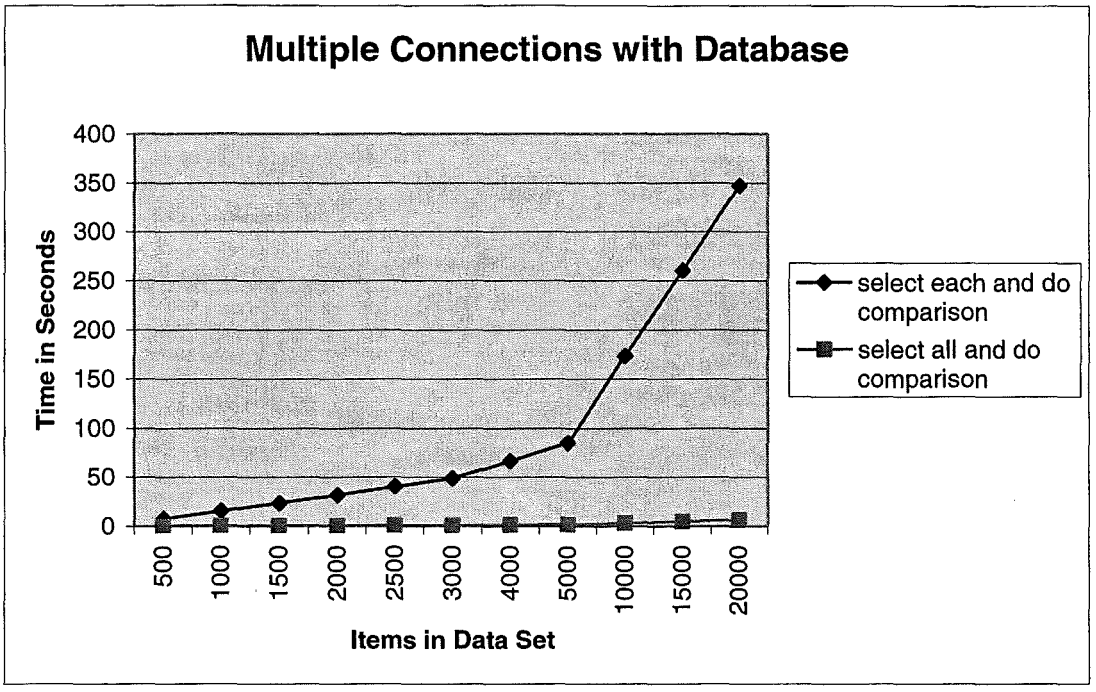


Figure 19: Comparison of Selecting All Data Sets and Performing Evaluations using Multiple Selects Versus a Single Select Statement During Multiple Connections to the Database

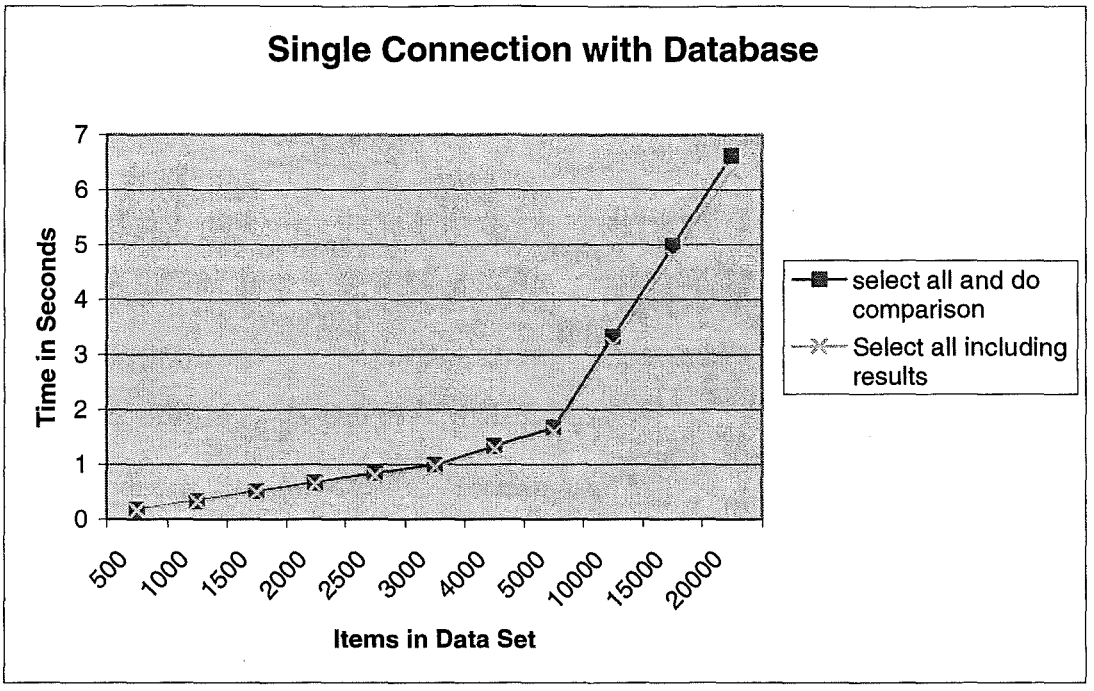


Figure 20: Comparison of Selecting All Data Sets and Performing Evaluations Versus Selecting All Data Sets Evaluated Previously During A Single Connection to the Database

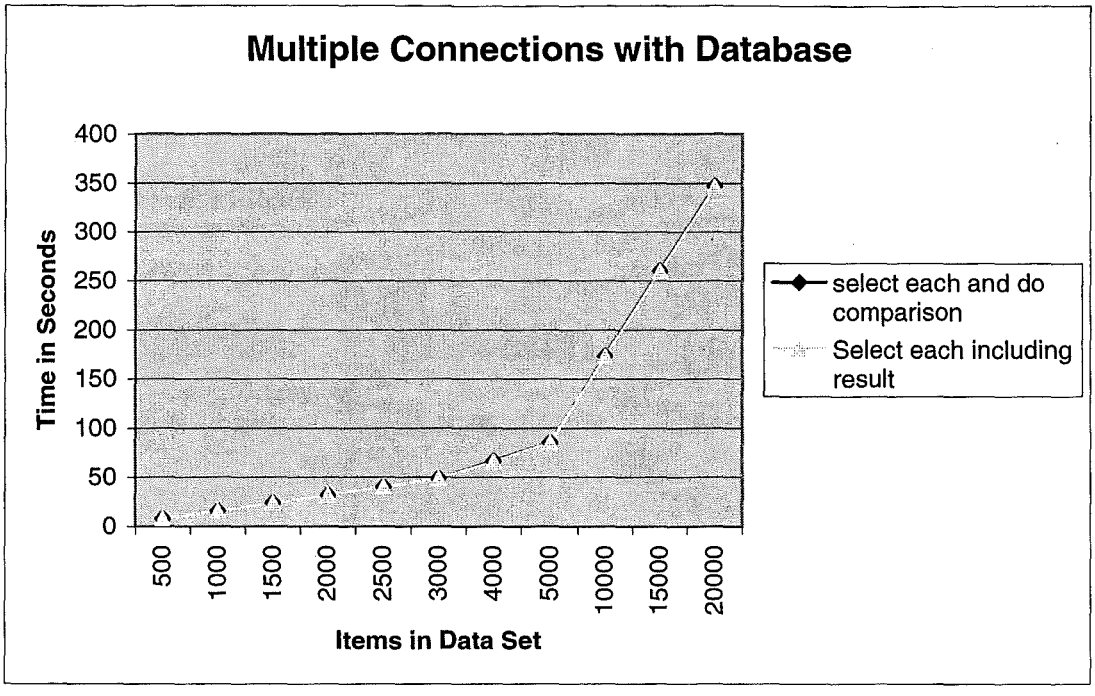


Figure 21: Comparison of Selecting All Data Sets and Performing Evaluations using Multiple Selects Versus a Single Select Statement of Data Sets Including Previously Evaluated Results During Multiple Connections to the Database

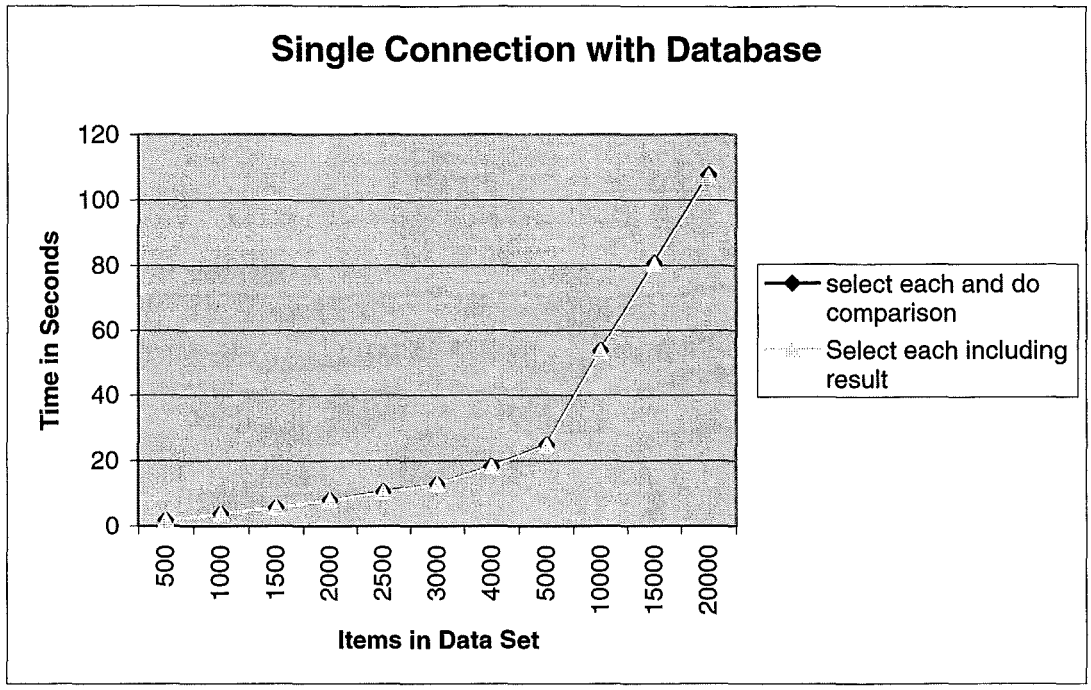


Figure 22: Comparison of Selecting All Data Sets and Performing Evaluations using Multiple Selects Versus a Single Select Statement of Data Sets Including Previously Evaluated Results During a Single Connection to the Database

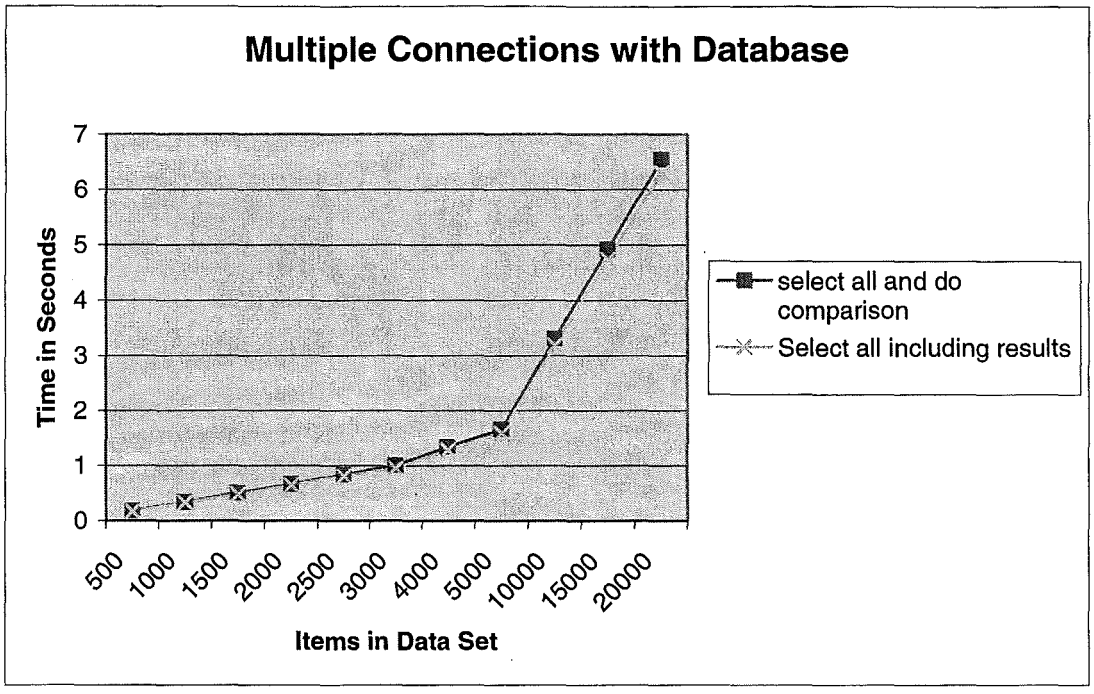


Figure 23: Comparison of Selecting All Data Sets in a Single Selecting Statement and Performing Evaluations Versus a Single Select Statement of Data Sets Including Previously Evaluated Results During Multiple Connections to the Database

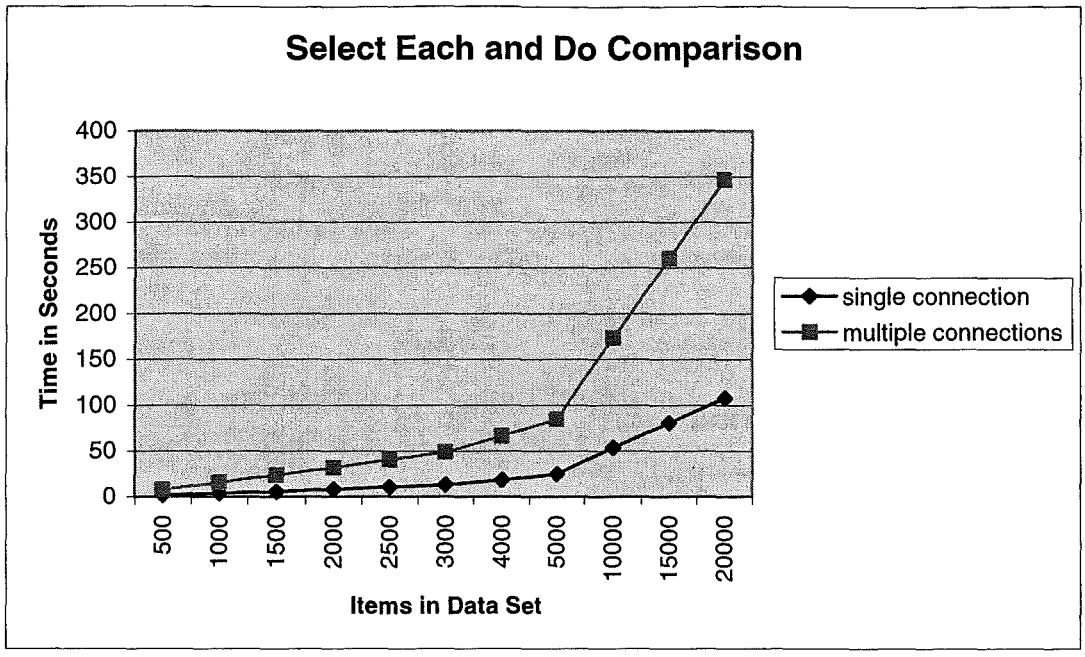


Figure 24: Comparison of Requesting Individual Data Sets with a Single Connection Versus Multiple Connections to the Database and then Performing Evaluations

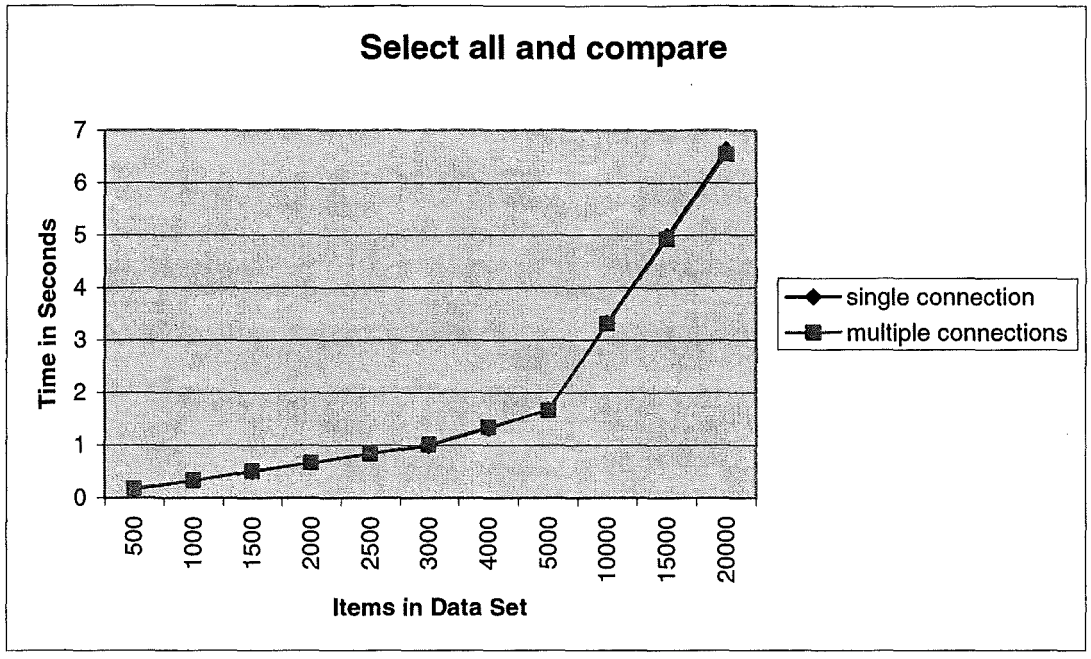


Figure 25: Comparison of Requesting All Data Sets with a Single Connection Versus Multiple Connections to the Database and then Performing Evaluations

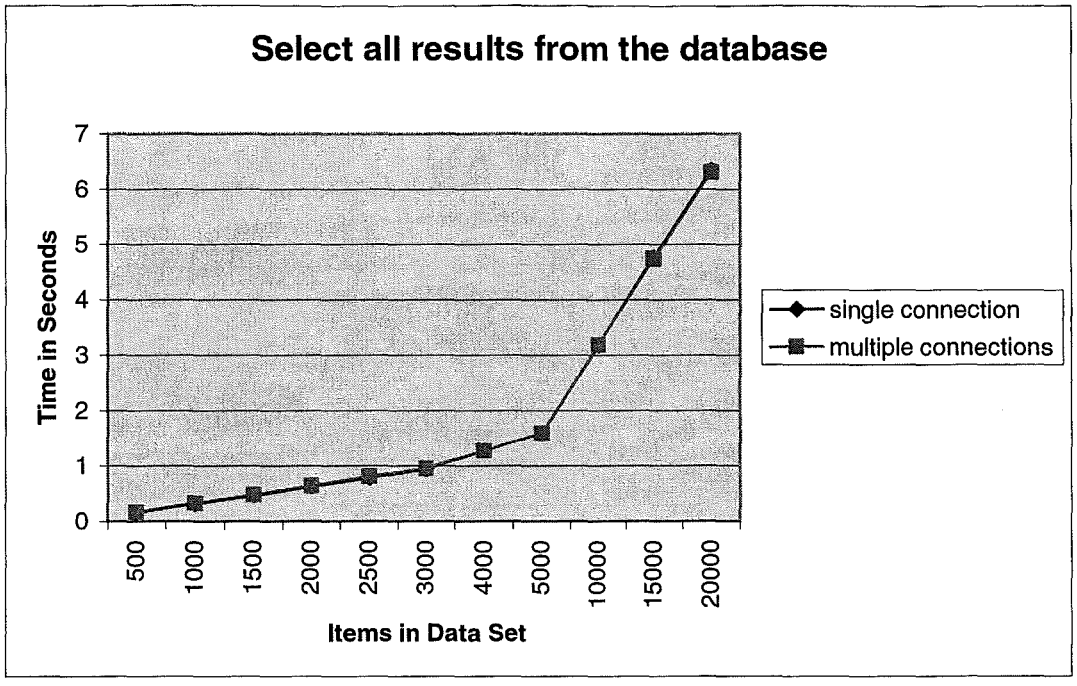


Figure 26: Comparison of Requesting All Data Sets including Previous Evaluated Results with a Single Connection Versus Multiple Connections to the Database

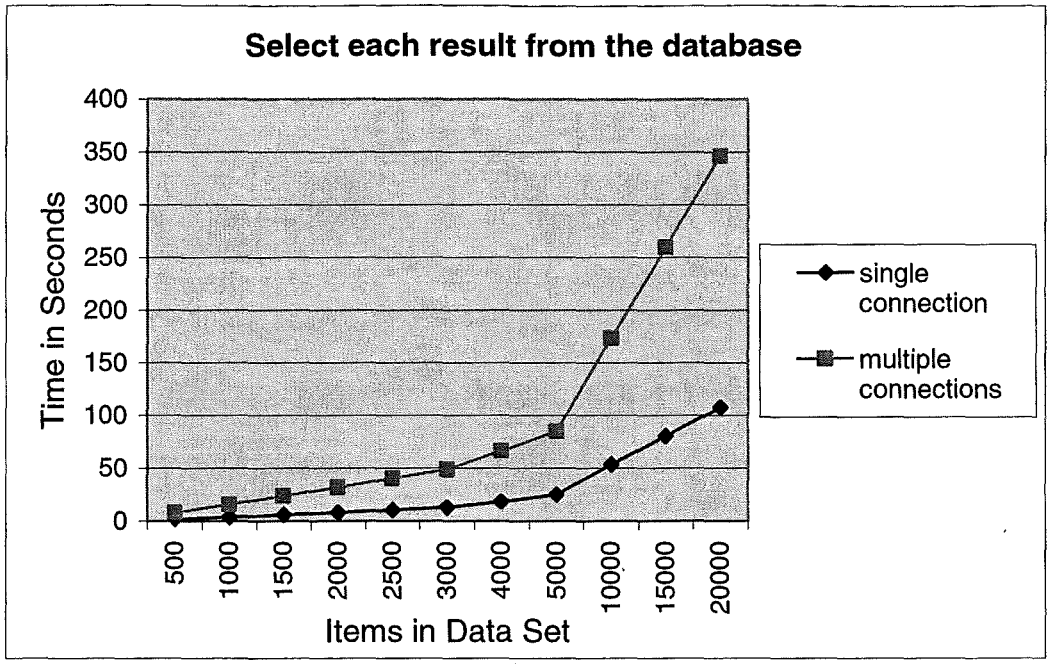


Figure 27: Comparison of Requesting Each Data Set Individually including Previous Evaluated Results with a Single Connection Versus Multiple Connections to the Database

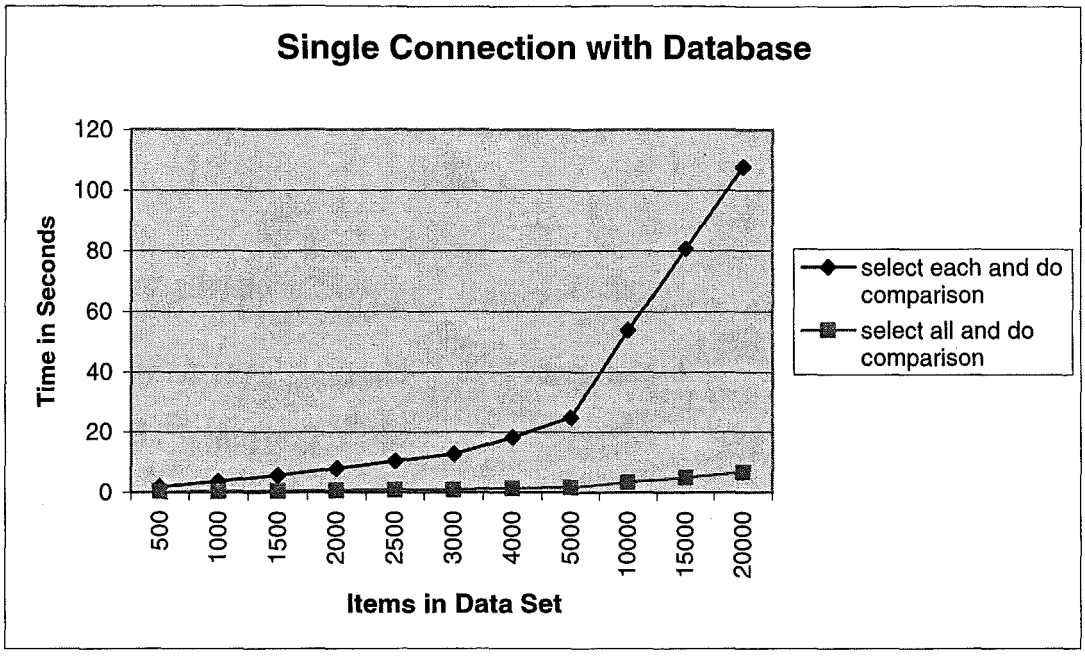


Figure 28: Comparison of Requesting Each Data Set Individually Versus Requesting All Data Sets with a Single Selection Statement Using a Single Connections to the Database and then Performing the Evaluation

APPENDIX I

Sample Date and Structure for Analysis

Source 1 - Table with five attributes

```
CREATE TABLE EMAILS (
  email varchar(70) NOT NULL default '',
  url varchar(70) NOT NULL default '',
  title varchar(70) NOT NULL default '',
  section int(1) NOT NULL default '0',
  timestamp timestamp(14) NOT NULL
) TYPE=MyISAM;
```

Source 1 - Instance Data Sample

```
bajan@dynamitemail.comb|http://hop.clickbank.net/hop.cgi?kwa
me/megapr|Get 5000 Hits To Your Website Per Hour!|1|2003-4-
9-9-9-15|

tele532002@yahoo.com|http://www.thisjob4you.tripod.com|Homeb
ased Telephone Workers Wanted|1|2003-4-9-9-35-52|

traker20012001@yahoo.com|http://inventorycontrol.resourcez.c
om/|Inventory Control for the Future.|1|2003-4-9-9-39-37|

huttonce@hotmail.com|http://beamentor.justrw.com/intro.php?i
d=Kris|Only $20 = $250 + $250 + $250 week after week|6|2003-
4-9-3-54-22|

sf1sf12003@yahoo.com|http://www.pairadicecasino.com|VEGAS
STYLE CASINO AND SPORTSBOOK|4|2003-4-9-23-53-30|

nadareply@yahoo.com|http://a1.lifetimebusiness.com|COULD YOU
USE ADDITIONAL INCOME?|1|2003-4-9-9-57-31|

marketing@wealthoptions.com|http://www.wealthoptions.com|Joi
n Our No B.S. Success Web Site.|5|2003-4-6-3-21-7|

aaaa@purpleone.freeserve.co.uk|http://www.kewlone.com|Dos
6.22/5.0/3.3 Win 3.1/203/101 WFW WP6 IE3/|2|2003-4-9-16-59-
25|
```

Source 2 - Table with fifteen attributes

```
CREATE TABLE areas (  areaID int(20) NOT NULL
auto_increment,  area_title varchar(60) default NULL,
area_desc varchar(255) default NULL,  dirpath varchar(255)
default NULL,  accessfile varchar(255) default NULL,
pwdfile varchar(255) default NULL,  templatefile
varchar(255) default NULL,  daily_price float(8,2) NOT NULL
default '0.00',  weekly_price float(8,2) NOT NULL default
'0.00',  monthly_price float(8,2) NOT NULL default '0.00',
quarterly_price float(8,2) NOT NULL default '0.00',
semi_annual_price float(8,2) NOT NULL default '0.00',
annual_price float(8,2) NOT NULL default '0.00',
one_time_price float(8,2) NOT NULL default '0.00',
default_pricing varchar(20) NOT NULL default
'one_time_price',  PRIMARY KEY  (areaID)) TYPE=MyISAM;
```

Source 2 - Sample of instance data

```
INSERT INTO areas (areaID, area_title, area_desc, dirpath,
accessfile, pwdfile, templatefile, daily_price,
weekly_price, monthly_price, quarterly_price,
semi_annual_price, annual_price, one_time_price,
default_pricing) VALUES (1, 'Free Trial', 'This is the
default area for the 15 day free trial membership.',
'/home/free9000/public_html/4you', '.htaccess', '.htpasswd',
'/home/freelead/public_html/manager/admin/htaccess.tpl',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'monthly_price');

INSERT INTO areas (areaID, area_title, area_desc, dirpath,
accessfile, pwdfile, templatefile, daily_price,
weekly_price, monthly_price, quarterly_price,
semi_annual_price, annual_price, one_time_price,
default_pricing) VALUES (2, '9000 Free Leads` Fast Pace
Membership', '9.95 basic membership',
'/home/freelead/public_html/fastpace', '.htaccess',
'.htpasswd',
'/home/freelead/public_html/manager/admin/htaccess.tpl',
'0.00', '0.00', '9.95', '0.00', '0.00', '0.00', '0.00',
'monthly_price');

INSERT INTO areas (areaID, area_title, area_desc, dirpath,
accessfile, pwdfile, templatefile, daily_price,
weekly_price, monthly_price, quarterly_price,
semi_annual_price, annual_price, one_time_price,
default_pricing) VALUES (3, '9000 Free Leads` Inner Circle
Membership', '$39.95 level',
'/home/free9000/public_html/innercircle', '.htaccess',
'.htpasswd',
'/home/freelead/public_html/manager/admin/htaccess.tpl',
'0.00', '0.00', '20.00', '0.00', '0.00', '0.00', '0.00',
'monthly_price');
```

```
INSERT INTO areas (areaID, area_title, area_desc, dirpath,
accessfile, pwdfile, templatefile, daily_price,
weekly_price, monthly_price, quarterly_price,
semi_annual_price, annual_price, one_time_price,
default_pricing) VALUES (4, '9000 Free Leads` POWER DRIVE
Membership (upgrade)', '$19.95 level',
'/home/freelead/public_html/upgrade', '.htaccess',
'.htpasswd',
'/home/freelead/public_html/manager/admin/htaccess.tpl',
'0.00', '0.00', '10.00', '0.00', '0.00', '0.00', '0.00',
'monthly_price');
```

Source 2 - Table with twenty attributes

```
CREATE TABLE profile (
  uid varchar(20) NOT NULL default '',
  realname varchar(60) default NULL,
  address varchar(60) default NULL,
  city varchar(40) default NULL,
  state varchar(40) default NULL,
  zip varchar(20) default NULL,
  country varchar(40) default NULL,
  phone varchar(40) default NULL,
  fax varchar(40) default NULL,
  email varchar(100) default NULL,
  aolim varchar(40) default NULL,
  icq varchar(30) default NULL,
  yahooim varchar(40) default NULL,
  msn varchar(40) default NULL,
  expires date default NULL,
  extra1 varchar(40) default NULL,
  extra2 varchar(40) default NULL,
  extra3 varchar(40) default NULL,
  extra4 varchar(40) default NULL,
  status varchar(20) default NULL
) TYPE=MyISAM;
```

Source 2 - Sample of instance data

```
INSERT INTO profile (uid, realname, address, city, state,
zip, country, phone, fax, email, aolim, icq, yahooim, msn,
expires, extra1, extra2, extra3, extra4, status) VALUES
('1', 'old username and password', '', '', '', '', '', '',
'', 'oldaccount@9000freeleads.net', '', '', '', '', '9999-
12-31', '', '', '', '', 'APPROVED');

INSERT INTO profile (uid, realname, address, city, state,
zip, country, phone, fax, email, aolim, icq, yahooim, msn,
expires, extra1, extra2, extra3, extra4, status) VALUES
('36', 'Christian Green', '2508 RT9W Oakbrook Manor 18N',
'Ravena', 'ny', '12143', 'usa', '', '', 'c_unique@msn.com',
'watchmeballout', '', 'nwo_cbanks', 'c_unique', '2002-12-
11', '', '', '', '', 'CANCELLED');
```

```
INSERT INTO profile (uid, realname, address, city, state,
zip, country, phone, fax, email, aolim, icq, yahooim, msn,
expires, extra1, extra2, extra3, extra4, status) VALUES
('35', 'Christian', '2508 RT9W Oakbrook Manor 18N',
'Ravena', 'ny', '12143', 'usa', '518.756.1990', '',
'c_unique@msn.com', 'watchmeballout', '', 'nwo_cbanks',
'c_unique', '2002-12-11', '', '', '', '', 'APPROVED');

INSERT INTO profile (uid, realname, address, city, state,
zip, country, phone, fax, email, aolim, icq, yahooim, msn,
expires, extra1, extra2, extra3, extra4, status) VALUES
('42', 'test', 'test', 'test', 'test', 'test', 'test', '',
'', 'john@marketingbots.com', '', '', '', '', '2003-02-19',
'', '', '', '', 'APPROVED');

INSERT INTO profile (uid, realname, address, city, state,
zip, country, phone, fax, email, aolim, icq, yahooim, msn,
expires, extra1, extra2, extra3, extra4, status) VALUES
('43', 'test', 'test', 'test', 'test', 'tst', 'test', '',
'', 'john@marketingbots.com', '', '', '', '', '2002-12-14',
'', '', '', '', 'CANCELLED');
```

APPENDIX J

DB Degrader Application

```
#############################################################
# program: Analyses program performance as large amounts
#  of data are loaded into the database
#
# section: Chapter 5 - Analysis
#
# description: Loads data sets of various sizes into
#  a database and then analyses the time perform a
comparison
#  of two item.  Analyses impact of database size on time
#  needed to complete the comparisons of the data items.
#############################################################

# load required modules used to access the db
# and perform time tests
use Time::HiRes qw( gettimeofday );
use DBI;

# clear output file
#open(FILE,">results.txt");
#close(FILE);

@file="";
# obtain data
open(FILE,"scripts.txt");
@file=<FILE>;
close(FILE);

# clear contents of table being used
$dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker");
$sql = "delete from test1";
$sth = $dbh->prepare($sql);
$sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";

# used to create the table the first run
#$dbh = DBI-
>connect("dbi:mysqlPP:database=thesis;host=localhost","jdant
e", "slacker");
```

```
#$sql = "create table test1 (field1 varchar(50), field2
varchar(50))";
#$sth = $dbh->prepare($sql);
#$sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";

# sort out the data to be used
foreach $line (@file){
     @each=split(/\,/, $line);
     push(@email,$each[4]);
     push(@username,$each[9]);
}
print "items split\n";

# records the size of the array holding the email addresses
$items=@email;

$memoryused=0;
$size=0;

# store analysis results to file
open(FILE,">>results.txt");

# clear memory structures
@email=();
@username=();
$memoryused=0;
@each=();

print FILE "- begin -\n";
# perform the analysis 50 times
for($j=0;$j<50;$j++){
                $sql = "insert into test1 values
(null,'$email[$i]','$items[$i]')";
                $sth = $dbh->prepare($sql);
                $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";

     $memoryused=$memoryused+length($email[$i])+length($items[$i
]);
          }
          $size=$size+$items;
     }

     # measure raw time doing comparison while in local
memory
     ($seconds1, $microseconds1) = gettimeofday;

     # conduct comparisons of random items in memory.
     # does this 20,000 times
     for($i=0;$i<20000;$i++){
          $chaos=int(rand($size));
#         print "inside loop $i, $chaos\n";
          # select each one and do comparison
```

```
            $sql = "select * from test1 where id = $chaos
limit 1";
            $sth = $dbh->prepare($sql);
            $sth->execute or print "Query failed - $sql -
\n($DBI::errstr)<br><br>";
            @instance = $sth->fetchrow;

            if($instance[0] =~ $instance[1]){
                $result[$i]=1;
            }else{
                $result[$i]=0;
            }
      }

      # measure raw time doing comparison while in local
memory
      ($seconds2, $microseconds2) = gettimeofday;

      # determine actual time needed for the comparisons
      $dif1 = $seconds2-$seconds1+(($microseconds2-
$microseconds1)/1000000);

      # store and display results
      print FILE "$j|$size|$memoryused|$dif1\n";
      print "$j|$size|$memoryused|$dif1\n";
}
```

APPENDIX K

Memory Degrader Application

```
#############################################################
# program: Analyses program performance as large amounts
#  of data are loaded into the local memory
#
# section: Chapter 5 - Analysis
#
# description: Loads data sets of various sizes into
#  a local memory and then analyses the time perform a
comparison
#  of two item.  Analyses impact of memory usage on time
#  needed to complete the comparisons of the data items.
#############################################################

# load required modules used to access the db
# and perform time tests
use Time::HiRes qw( gettimeofday );
print "Content-type: text/html\n\n";

# clear output file
#open(FILE,">results.txt");

@file="";
# obtain data
open(FILE,"scripts.txt");
@file=<FILE>;
close(FILE);

# clear memory locations needed
$memoryused=0;
@email=();
@username=();
$memoryused=0;
@each=();

open(FILE,">>results.txt");
print FILE "\n\n- begin -\n";
close FILE;

# perform the analysis 50 times
for($j=0;$j<60;$j++){
     # load the data from file into memory 20 times
     for($i=0;$i<20;$i++){
```

```
foreach $line (@file){
                @each=split(/\,/, $line);
                push(@email,$each[4]);
                push(@username,$each[9]);

     $memoryused=$memoryused+length($each[4])+length($each[9
]);
          }
     }
     $size=@email;

     # measure raw time doing comparison while in local
memory
     ($seconds1, $microseconds1) = gettimeofday;

     # conduct comparisons of random items in memory.
     # does this 20,000 times
     for($i=0;$i<20000;$i++){
          $chaos=int(rand($size));
          if($email[$chaos] =~ $username[$chaos]){
               $result[$chaos]=1;
          }else{
               $result[$chaos]=0;
          }
     }

     # measure raw time doing comparison while in local
memory
     ($seconds2, $microseconds2) = gettimeofday;

     # determine actual time needed for the comparisons
     $dif1 = $seconds2-$seconds1+(($microseconds2-
$microseconds1)/1000000);

     # store and display results
     open(FILE,">>results.txt");
     print FILE "$j|$size|$memoryused|$dif1\n";
     print "$j|$size|$memoryused|$dif1\n";
     close(FILE);
}
```

References

[Beneventano01]
Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M., "The MOMIS approach to Information Integration", IEEE and AAAI International Conference on Enterprise Information Systems, Setúbal, Portugal, 7-10 July, 2001.

[Beneventano98]
Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.,"MOMIS: An Intelligent System for the Integration of Semistructured and Structured Data", T3-R07, Universita degli Studi di Modena e Reggio Emilia, 10 November 1998.

[Chapple03]
Chapple, Mike, 'Databases', About.com, http://databases.about.com/library/glossary/bldef-metadata.htm, 2003.

[Clifton97]
Clifton, Chris, Housman, Ed, and Rosenthal, Arnon, "Experience with a Combined Approach to Attribute-Matching Across Heterogeneous Databases", IEEE Workshop on Data Semantics, Leysin, Switzerland, 1997, p 1-24.

[Cohen98]
Cohen, William W, "Integration of Heterogeneous Databases Without Common Domains Using Query Based on Textual Similarity", Proceedings of the 1998 ACM SIGMOD. Seattle, WA, 1998.

[Doan00]
Doan, AnHai, Domingos, Pedro, and Levy, Alon, "Learning Source Descriptions for Data Integration", Proceedings of WehDB 2000, Dallas, TX. 2000.

[Gelati01]
Gelati, G., Guerra, F., Vincini, M., "Agents Supporting Information Integration: the MIKS Framework", Workshop Dagli Oggetti Agli Agenti, Modena, Italy, September 4 - 5, 2001.

[Geist01]
Geist, Ingolf, Sattler, Kai-Uwe, and Schmitt, Ingo, "Combining a Formal with an Example-drive Approach for Data Integration", Foderierte Datenbanken. 2001. p 30-48.

[Goidman99]
Goidman, Roy, and Widom, Jennifer, "Approximate DataGuides", Proceedings of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats, Jerusalem, Israel, January 1999.

[Madhavan01]
Madhavan, Jayant, Bernstein, Philip A., and Rahm, Erhard, "Generic Schema Mapping with Cupid", Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01), Rome, Italy, September 2001, pp. 49-58.

[Miller01]
Miller, Renee J., Hernandez, Mauricio A., Haas, Laura M., Yan, Lingling, Ho, C. T. Howard, Fagin, Ronald, and Popa, Lucian, "The Clio Project: Managing Heterogeneity", ACM SIGMOD Record 30, 1 (March 2001), pp. 78-83.

[Mitra99]
Mitra, Prasenjit, Wiederhold, Gio, and Jannink, Jan, "Semi-Automatic Integration of Knowledge Sources", Proceedings of Fusion '99, Sunnyvale CA, July 1999.

[Rahm01]
Rahm, Erhard, and Bernstein, Philip A., "On Matching Schemas Automatically", VLDB Journal, 10, 4 (Dec. 2001).

[Russell95]
Russell, S., and Norvig, P., Artificial Intelligence: A Modern Approach, Prentice Hall, Saddle River, NJ, USA, 1995, p. 97-101.

[Tannenbaum02]
Tannenbaum, Adrienne, Metadata Solutions: Using Metamodels, Repositories, XML, and Enterprise Portals to Generate Information on Demand, Addison-Wesley, Boston, MA, USA, 2002, p. 93, 100-156.

[Webmaster03]
Webmaster@linux.org, "What is Linux", Linux.org, http://www.linux.org/info/index.html, 2003.

VITA

John S. Dembowski was born in , and was raised in San Diego and Lemoore, California, and Jacksonville, Florida. He has lived in Atlanta, GA; Albuquerque, NM; and Subic Bay, Philippines. He held positions as a Marketing Manager and a System Automater for AT&T Universal Card. Previous business ventures include two nightclubs; an international music festival for the new millennium produced in Mexico, organized over the Internet, and sponsored by over 100 U.S. based radio stations; and a small music publication. He is the author of several dozen commercially available automated marketing software suites. He is the owner of MegaScorpion Recordings, an Internet-based music label with a roster of over 100 international artists, and embraces Peer2Peer, MP3s, and the Internet as a distribution channel. He is the front man for the electronic band Late Night Adventurers and appeared on 35 internationally released compact discs during his last year in college. Currently he develops network applications, owns and manages twenty websites on a variety of topics, and consults for clients in several cities across the United States.

He received a Bachelor and Master of Science in Computer and Information Sciences from the University of North Florida. He prefers to program in Perl but also writes in C, Java, and PHP. Previous software experience includes ASP, JSP, COBOL, Visual Basic, Basic, Pascal, SAS, SPSS, and Prolog. Database experience includes work with Oracle and MySQL and covers DBA activities, data mining, data warehousing, and disaster recovery.